



**ITS**  
Institut  
Teknologi  
Sepuluh Nopember

**TUGAS AKHIR - TE145561**

**MONITORING DAN KONTROL POMPA AIR PADA  
RUMAH KABEL BAWAH TANAH**

Devi Maulina  
NRP 2212 038 007  
Andi Suhendra Prayuda  
NRP 2212 038 009

Dosen Pembimbing  
Ir. Josaphat Pramudijanto, M. Eng.

PROGRAM STUDI D3 TEKNIK ELEKTRO  
Fakultas Teknologi Industri  
Institut Teknologi Sepuluh Nopember  
Surabaya 2015



**ITS**  
Institut  
Teknologi  
Sepuluh Nopember

*FINAL PROJECT - TE145561*

*MONITORING AND WATER PUMP CONTROL OF  
UNDERGROUND CABLE HOUSE*

Devi Maulina  
ID 2212038007  
Andi Suhendra Prayuda  
ID 2212038009

*Supervisor*  
Ir. Josaphat Pramudijanto, M. Eng.

*ELECTRICAL ENGINEERING D3 STUDY PROGRAM  
Industrial Technology Faculty  
Institut of Technolgy Sepuluh Nopember  
Surabaya 2015*

# **MONITORING DAN KONTROL POMPA AIR PADA RUMAH KABEL BAWAH TANAH**

## **TUGAS AKHIR**

**Diajukan Guna Memenuhi Sebagian Persyaratan  
Untuk Memperoleh Gelar Ahli Madya  
Pada  
Bidang Studi Teknik Listrik  
Program Studi D3 Teknik Elektro  
Fakultas Teknologi Industri  
Institut Teknologi Sepuluh Nopember**

**Menyetujui :  
Dosen Pembimbing,**



**Ir. Josaphat Pramudijanto, M.Eng  
NIP. 19621005 199003 1 003**

**SURABAYA  
JULI, 2015**

## **MONITORING DAN KONTROL POMPA AIR PADA RUMAH KABEL BAWAH TANAH**

**Nama Mahasiswa 1** : **Devi Maulina**  
**NRP** : **2212 038 007**  
**Nama Mahasiswa 2** : **Andi Suhendra Prayuda**  
**NRP** : **2212 038 009**  
**Dosen Pembimbing** : **Ir. Josaphat Pramudijanto, M. Eng.**  
**NIP** : **19621005 199003 1003**

### **ABSTRAK**

Rumah kabel bawah tanah adalah suatu ruangan dimana didalamnya terdapat beberapa kabel listrik tegangan menengah pada instalasi gardu induk PLN. Pada beberapa gardu induk di Surabaya terdapat beberapa kabel yang isolasinya sudah rusak. Kerusakan pada isolasi kabel ini biasanya dikarenakan oleh rendaman air pada kabel dengan waktu yang lama. Rendaman air ini berasal dari air hujan maupun air limbah pada selokan disekitar gardu induk yang datarannya lebih tinggi sehingga menimbulkan rembesan air ke gardu induk khususnya pada rumah kabel. Dengan adanya hal ini, petugas dituntut untuk selalu tanggap dalam mengawasi keadaan rumah kabel terhadap rembesan air yang bisa membanjiri tempat tersebut dan memberi informasi kapan waktunya air harus dibuang. Cara pelaporan ini terbilang kurang efektif dan efisien.

Alat ini mampu mengukur ketinggian air secara *real time*, dan mampu mengontrol ketinggian air. Jika tinggi air lebih dari 13 cm maka pompa 2 akan *on*, dan pompa 2 akan *off* apabila ketinggian air yang semula lebih dari 13 cm menjadi kurang dari 3 cm. Kemudian pada pompa 1 *on* apabila air menggerakkan sensor pada tandon dengan ditandai status sensor tandon *on*. Pembacaan ketinggian air dan status pompa air akan ditampilkan pada LCD dan PC melalui *software* LabVIEW dan disimpan pada *database* Ms.Excel PC.

Dari hasil Tugas Akhir ini dapat memonitor ketinggian air dan kerja pompa air *on off* secara *real time* melalui PC maupun pada LCD panel monitoring. Pada PC data tersimpan secara otomatis selang 1 menit ketika ketinggian air dibawah 10 cm dan ketika diatas 10 cm (kritis) data tersimpan selang 30 detik kemudian dari hasil itu di *download* pada LabVIEW untuk dijadikan *database*.

**Kata Kunci** : Monitoring, Kontrol Pompa Air, Sensor Ketinggian Air.

*Halaman ini sengaja dikosongkan*

## **MONITORING AND WATER PUMP CONTROL OF UNDERGROUND CABLE HOUSE**

**Student Name 1** : **Devi Maulina**  
**ID** : **2212 038 007**  
**Student Name 2** : **Andi Suhendra Prayuda**  
**ID** : **2212 038 009**  
**Supervisor** : **Ir. Josaphat Pramudijanto, M. Eng.**  
**ID** : **19621005 199003 1003**

### **ABSTRACT**

*Underground cable house is a room in which there are several medium voltage power cables the installation of substations PLN. In some substations in Surabaya, there are several cable insulation is damaged. Damage to the cable insulation is usually caused by a water immersion at a cable with a long time. This water immersion comes from rain water and waste water in sewers around substations that higher plains causing water seepage into the substation, especially in cable house. With the existence of this, officers are required to be responsive to monitor the state of the cable house to water seepage that could overwhelm the place it and provide information when to water should be discThe way of reporting is somewhat less effective and efficient .*

*The prototype is capable to measure the water level in real time, and it is able to control the water level. If the condition of water level is upper 13 cm, then pump 2 will be on, and pump 2 will be off if the water level were originally more than 13 cm became less than 3 cm. Then the pump 1 on when the water moves sensor on the tank with tank sensor status marked on. The reading of the water level and the water pump status is displayed on the LCD and PC through LabVIEW software and will be saved on a database Ms.Excel PC*

*From the results of this final project can monitor the water level and water pump working on off in real time via the PC or on the LCD panel monitoring. Data will be saved on the PC automatically lapse 1 minute when the water level below 10 cm and when over 10 cm (critical) data will be saved interval of 30 seconds later from the results downloaded in LabVIEW to be used as database.*

**Keywords:** *Monitoring, pump control, water level sensor*

*Halaman ini sengaja dikosongkan*

## **KATA PENGANTAR**

Dengan mengucapkan puji syukur kehadirat Allah SWT yang telah melimpahkan rahmat, hidayah serta karunia-Nya sehingga penulis dapat menyelesaikan Tugas Akhir dengan judul :

### **“MONITORING DAN KONTROL POMPA AIR PADA RUMAH KABEL BAWAH TANAH”**

Tugas akhir ini merupakan sebagian syarat untuk menyelesaikan mata kuliah dan memperoleh nilai pada tugas akhir.

Dengan selesainya tugas akhir ini penulis menyampaikan terima kasih sebesar-besarnya kepada:

1. Orang Tua atas limpahan doa, kasih sayang, dukungan dan dorongan baik berupa moril atau materil bagi penulis.
2. Bapak Eko Setijadi, ST. MT. Ph.D selaku Ketua Program D3 Teknik Elektro Bidang Studi Teknik Listrik, FTI-ITS Surabaya.
3. Bapak Ir. Josaphat Pramudijanto, M.Eng. selaku dosen pembimbing.
4. Seluruh staf pengajar dan administrasi Prodi D3 Teknik Elektro FTI-ITS.
5. Seluruh mahasiswa D3 Teknik Elektro khususnya angkatan 2012.
6. Semua pihak yang telah banyak membantu untuk menyelesaikan tugas akhir ini yang tidak dapat penulis sebutkan satu persatu.

Kritik dan saran untuk perbaikan Tugas Akhir ini sangat diperlukan. Akhir kata semoga Tugas Akhir ini dapat bermanfaat bagi kita semua.

Surabaya, Juli 2015

Penulis



*Halaman ini sengaja dikosongkan*

## DAFTAR ISI

HALAMAN JUDUL .....	i
LEMBAR PENGESAHAN .....	iii
ABSTRAK .....	v
<i>ABSTRACT</i> .....	vii
KATA PENGANTAR.....	ix
DAFTAR ISI .....	xi
DAFTAR GAMBAR .....	xiii
DAFTAR TABEL .....	xv

### BAB I PENDAHULUAN

1.1 Latar Belakang .....	1
1.2 Permasalahan.....	1
1.3 Batasan Masalah .....	1
1.4 Tujuan .....	2
1.5 Sistematika Laporan.....	2
1.6 Relevansi .....	3

### BAB II TEORI PENUNJANG

2.1. Rumah Kabel Bawah Tanah .....	5
2.2. Pompa .....	6
2.2.1 Pompa Air Otomatis .....	6
2.3 Sensor Ultrasonik HY-SRF05 .....	7
2.4. <i>Liquid Water Level Sensor Right Angle Float Switch</i> .....	8
2.5 Arduino Mega .....	8
2.6 Arduino <i>Shield Ethernet</i> .....	9
2.7 LCD .....	10
2.8 Modul I2C LCD <i>Backpak</i> .....	11
2.9 Komunikasi <i>Serial Kabel Ethernet (RJ45)</i> .....	12
2.10 <i>LabVIEW</i> .....	14

### BAB. III PERANCANGAN DAN PEMBUATAN ALAT

3.1 Perancangan Keseluruhan .....	17
3.2 Perancangan Mekanik .....	18
3.3 Perancangan <i>Hardware</i> .....	20
3.3.1 Perancangan Sensor Ultrasonik HY-SRF05.. .....	20
3.3.2 Perancangan <i>Liquid Water Level Sensor Right Angle Float Switch</i> .....	21

3.3.3	Perancangan LCD.....	21
3.3.4	Perancangan Modul Arduino Mega.....	22
3.3.5	Perancangan Modul Relai .....	23
3.4	Perancangan <i>Software</i> .....	24
3.4.1	Program Pada Arduino Mega .....	25
3.4.2	<i>LabVIEW</i> .....	27
3.4.3	Tampilan Monitoring .....	27
 <b>BAB. IV PENGUJIAN DAN ANALISA DATA</b>		
4.1	Pengujian Arduino Mega.....	32
4.2	Pengujian Relai.....	32
4.3	Pengujian Ultrasonik HY-SRF05 .....	33
4.4	Pengujian <i>Liquid Water Level Sensor Right Angle Float Switch</i> .....	36
4.5	Pengujian Komunikasi dengan Kabel <i>Ethernet</i> .....	37
4.6	Pengujian HMI .....	38
4.6.1	Pengujian Tampilan Monitoring.....	38
4.6.2	Pengujian <i>Database</i> .....	40
4.7	Pengujian Alat Keseluruhan .....	42
 <b>BAB. V PENUTUP</b>		
5.1.	Kesimpulan .....	45
5.2.	Saran .....	45
 <b>DAFTAR PUSTAKA .....</b>		
<b>Lampiran 1</b> Skema Purwarupa Rumah Kabel .....		A-1
<b>Lampiran 2</b> <i>Listing</i> Program pada Arduino .....		B-1
<b>Lampiran 3</b> Foto Alat Hasil Pembuatan Purwarupa.....		C-1
<b>Lampiran 4</b> <i>Data Sheet</i> .....		D-1
<b>RIWAYAT HIDUP PENULIS .....</b>		E-1

## DAFTAR TABEL

<b>Tabel 2.1</b>	Konfigurasi I2C LCD <i>Backpack</i> pada Arduino.....	12
<b>Tabel 4.1</b>	Hasil Pengukuran Sensor Ultrasonik HY-SRF05.....	35

*Halaman ini sengaja dikosongkan*

## DAFTAR GAMBAR

<b>Gambar 2.1</b>	Rumah Kabel Bawah Tanah .....	5
<b>Gambar 2.2</b>	Rangkaian Listrik yang Terdapat pada Sebuah Pompa Air .....	7
<b>Gambar 2.3</b>	Ultrasonik HY-SRF05 .....	8
<b>Gambar 2.4</b>	Arduino Mega .....	9
<b>Gambar 2.5</b>	<i>Shield Ethernet</i> .....	10
<b>Gambar 2.6</b>	Diagram Pin LCD .....	11
<b>Gambar 2.7</b>	Modul I2C LCD <i>Backpack</i> .....	11
<b>Gambar 2.8</b>	Kabel UTP dengan standar T-568B dan T-568B serta konektor RJ 45 .....	12
<b>Gambar 2.9</b>	Kabel untuk <i>Straigh Through</i> .....	13
<b>Gambar 2.10</b>	Kabel untuk <i>Cross Over</i> .....	13
<b>Gambar 2.11</b>	<i>Front Panel LabVIEW</i> .....	14
<b>Gambar 2.12</b>	Blok Diagram <i>LabVIEW</i> .....	15
<b>Gambar 2.13</b>	<i>Control Pallette</i> Pada <i>Lab Front Panel LabVIEW</i> .....	15
<b>Gambar 2.14</b>	Tampilan <i>Functions Pallette</i> .....	16
<b>Gambar 3.1</b>	Diagram Fungsional Sistem Alat Secara Keseluruhan .....	17
<b>Gambar 3.2</b>	Purwarupa Rumah Kabel .....	19
<b>Gambar 3.3</b>	Konfigurasi Pin Sensor Ultrasonik HY-SRF05 .....	20
<b>Gambar 3.4</b>	Konfigurasi Pin <i>Liquid Water Level Sensor Right Angle Float Switch</i> .....	21
<b>Gambar 3.5</b>	Konfigurasi Pin LCD .....	22
<b>Gambar 3.6</b>	Penggunaan Pin Pada Arduino Mega .....	23
<b>Gambar 3.7</b>	<i>Driver Relai</i> .....	24
<b>Gambar 3.8</b>	<i>Flow Chart</i> Arduino .....	26
<b>Gambar 3.9</b>	Tampilan Form Utama .....	27
<b>Gambar 3.10</b>	<i>Flow Chart LabVIEW</i> .....	29
<b>Gambar 3.11</b>	Lanjutan <i>Flow Chart LabVIEW</i> .....	30
<b>Gambar 4.1</b>	Hasil Pembuatan <i>Hardware</i> Secara Keseluruhan .....	31
<b>Gambar 4.2</b>	Pengujian Arduino .....	32
<b>Gambar 4.3</b>	Pengujian Rangkaian Relai .....	33
<b>Gambar 4.4</b>	Hasil Pengujian Relai .....	33
<b>Gambar 4.5</b>	Pengujian Sensor Ultrasonik HY-SRF05 .....	34
<b>Gambar 4.6</b>	Kurva Pengujian Sensor HY-SRF05 .....	35
<b>Gambar 4.7</b>	Keadaan <i>Switch Closed</i> .....	36
<b>Gambar 4.8</b>	Keadaan <i>Switch Open</i> .....	37
<b>Gambar 4.9</b>	Pengujian Koneksi .....	37
<b>Gambar 4.10</b>	Hasil <i>Ping Test</i> .....	38

<b>Gambar 4.11</b>	Blok Diagram <i>host IP</i> .....	39
<b>Gambar 4.12</b>	Tampilan Utama Monitoring .....	39
<b>Gambar 4.13</b>	Tampilan HMI untuk <i>Admin Database</i> .....	40
<b>Gambar 4.14</b>	Tampilan Untuk <i>Download Database</i> .....	41
<b>Gambar 4.15</b>	Tampilan <i>Database</i> dalam <i>Ms.Excel</i> .....	41
<b>Gambar 4.16</b>	Tampilan Awal LCD .....	42
<b>Gambar 4.17</b>	Kondisi Ketinggian 8 cm .....	43
<b>Gambar 4.18</b>	Kondisi Kedua Ketinggian Diatas 13 cm .....	43

# **BAB I**

## **PENDAHULUAN**

### **1.1 Latar Belakang**

Pada daerah yang berada dibawah permukaan air laut seperti Surabaya sering sekali terjadi banjir yang biasanya terjadi pada musim hujan dan membuat air menggenangi beberapa tempat di Surabaya dan tak kunjung surut. Hal semacam ini, kami melihat pada suatu tempat pada gardu induk distribusi yang terdapat semacam rumah kabel bawah tanah yang terdapat kabel tanah didalamnya. Masalah yang terdapat pada kabel tanah ini secara umum adalah air menggenangi rumah kabel bawah tanah dan kabel terendam oleh air yang terlalu lama yang diakibatkan oleh hujan. Hal tersebut juga dikarenakan pada dasar rumah kabel bawah tanah bidang tanahnya datar yang menyebabkan tidak adanya aliran air yang membuat air menuju satu titik pembuangan.

Meskipun pada kabel terdapat isolator pembungkus, apabila terendam air dalam waktu yang lama akan terjadi rembesan kedalam inti/tembaga kabel, apabila hal tersebut terjadi potensi aliran listrik mengalir pada air menjadi besar. Kemudian digunakan sensor ketinggian air yang nantinya sensor ini sebagai sistem yang memonitoring kondisi ketinggian air dalam rumah kabel bawah tanah dan memberikan respon sebagai usaha mengontrol ketinggian air tersebut dengan menggunakan pompa air dengan memompa air keluar dari rumah kabel bawah tanah.

### **1.2 Permasalahan**

Beberapa permasalahan yang ada antara lain :

- a. Kurangnya pengawasan yang mendetail pada rumah kabel bawah tanah perihal tentang ketinggian air yang dapat merendam kabel.
- b. Pada rumah kabel selalu tergenang oleh air dan rawan banjir pada saat hujan.
- c. Untuk memompa rumah kabel masih menggunakan pompa secara manual.

### **1.3 Batasan Masalah**

Dari perumusan masalah di atas, maka batasan masalah dari pembuatan Tugas Akhir ini adalah :

- a. Alat monitoring dan kontrol ini berupa purwarupa yang dibuat untuk mensimulasikan ketinggian air dan kerja pompa air pada rumah kabel bawah tanah.



- b. Pengiriman data hasil monitoring menggunakan kabel *Ethernet* sebagai media komunikasi antara *hardware* dan komputer *server*.

#### **1.4 Tujuan**

Tujuan yang akan dicapai dalam kegiatan Tugas Akhir ini diantaranya adalah:

- a. Merancang dan membuat sistem pengiriman data dari kondisi air pada rumah kabel bawah tanah menggunakan Sensor Level Air yang kemudian dikirimkan menggunakan arduino melalui kabel *Ethernet*.
- b. Membuat purwarupa sistem monitoring dan kontrol pompa air rumah kabel bawah tanah.
- c. Membuat *Interface* berupa tampilan atau monitoring yang dibuat dengan bahasa pemrograman *LabVIEW* dan program yang mendukung.

#### **1.5 Sistematika Laporan**

Pembahasan pada laporan Tugas Akhir ini terdiri dari lima bab. Berikut diberikan penjelasan isi dari masing – masing bab, yaitu

##### **BAB I : PENDAHULUAN**

Dalam bab ini membahas tentang latar belakang, permasalahan, batasan masalah, maksud dan tujuan, sistematika laporan, serta relevansi penulisan pada Tugas Akhir.

##### **BAB II : TEORI PENUNJANG**

Berisi teori penunjang yang mendukung dalam perencanaan dan pembuatan alat.

##### **BAB III: PERANCANGAN DAN PEMBUATAN ALAT**

Perancangan dan pembuatan alat meliputi tentang perencanaan purwarupa sistem monitoring dan kontrol, pembuatan *hardware* yang meliputi rangkaian elektrik, purwarupa, dan perancangan *software* yang meliputi program diantaranya program Arduino dan *LabVIEW*.

##### **BAB IV : PENGUJIAN DAN ANALISA DATA**

Membahas tentang pengukuran, pengujian, dan penganalisaan terhadap kepresisian sensor dan purwarupa yang telah kami buat .

## **BAB V : PENUTUP**

Menjelaskan tentang kesimpulan dan saran dari Tugas Akhir untuk pengembangan purwarupa ini lebih lanjut.

### **1.6 Relevansi**

Tujuan yang ingin kami capai adalah purwarupa alat monitoring dan kontrol pompa air pada rumah kabel bawah tanah ini dapat direalisasikan menjadi alat monitoring kinerja pompa air sungguhan. Bermanfaat untuk PT PLN (Persero) terutama pada gardu induk yang sering terjadi banjir. Guna meringankan tugas operator dalam menjalankan dan memonitoring proses pompa air yang sedang bekerja dan memudahkan operator untuk memperoleh data ketinggian air pada rumah kabel bawah tanah yang akurat. Dari data tersebut nantinya dapat digunakan untuk perbaikan gardu induk kedepannya.

*Halaman ini sengaja dikosongkan*

## **BAB II**

### **TEORI PENUNJANG**

Pada bab ini membahas tentang teori dasar dan teori penunjang dari peralatan-peralatan yang digunakan dalam pembuatan alat monitoring ketinggian air.

#### **2.1 Rumah Kabel Bawah Tanah [1]**

Kabel tanah adalah salah satu / beberapa kawat yang diisolasikan, sehingga tahan terhadap tegangan tertentu antara penghantar yang satu dengan penghantar yang lain ataupun penghantar dengan tanah serta dibungkus dengan pelindung, sehingga terhindar dari pengaruh kimia lain yang ada dalam tanah. Dikarenakan kabel tanah tersebut beroperasi dalam tanah, maka komponen termasuk kabel harus mampu beroperasi secara terus menerus karena memiliki persyaratan isolasi yang khusus untuk melindunginya dari segala bentuk kelembaban serta pengaruh pengaruh lain yang terdapat didalam tanah. Untuk rumah kabel bawah tanah pada gardu induk dapat dilihat pada Gambar 2.1



**Gambar 2.1** Rumah Kabel Bawah Tanah

Suatu kabel tenaga dapat tersusun dari beberapa lapisan penghantar dan isolasi. Isolasi pada kabel tenaga merupakan hal yang sangat penting diperhatikan agar terhindar dari kerusakan yang berakibat pada kebocoran arus pada bahan isolasi. Isolasi berfungsi untuk memisahkan bagian-bagian yang mempunyai beda tegangan supaya di antara bagian-bagian tersebut tidak terjadi lompatan listrik atau percikan bunga api.

## 2.2 Pompa [2]

Pompa adalah alat yang digunakan untuk memindahkan cairan (fluida) dari suatu tempat ke tempat yang lain, melalui media pipa (saluran) dengan cara menambahkan energi pada cairan yang dipindahkan dan berlangsung terus menerus.

Pompa beroperasi dengan prinsip membuat perbedaan tekanan antara bagian hisap (*suction*) dan bagian tekan (*discharge*). Dengan kata lain, pompa berfungsi mengubah tenaga mekanis dari suatu sumber tenaga (penggerak) menjadi tenaga kinetis (kecepatan), dimana tenaga ini berguna untuk mengalirkan cairan dan mengatasi hambatan yang ada sepanjang pengaliran.

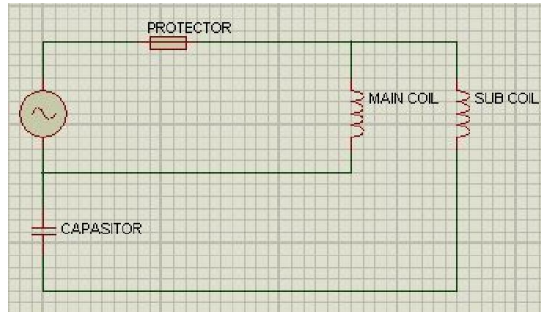
### 2.2.1 Pompa Air Otomatis [2]

Pada dasarnya setiap pompa air dilengkapi dengan peralatan otomatis ketika kita membeli mesin pompa air di toko, ini berguna untuk memudahkan kita pada saat pengoperasian, sehingga waktu kita menjadi lebih efektif dan efisien dan tidak memerlukan aktifitas menghidupkan ataupun mematikan pompa, sebab sudah ada sensor otomatisnya, yang bekerja berdasarkan tekanan yang terdapat pada pipa tau saluran air pada keluaran pompa.

Pada mesin pompa air ada saluran hisap dan ada saluran buang, alat otomatis atau sensornya menggunakan sensor tekanan atau disebut juga *Pressure Switch* dan dipasang pada tabung pada saluran keluaran pompa, ketika pompa dihidupkan atau dihubungkan dengan tegangan jala-jala, maka pompa akan berputar sehingga dibagian dalam pompa terjadi *vaccum* karena adanya perbedaan tekanan, sehingga air yang ada didalam tanah akan terhisap naik.

Pada saat mesin pompa air berputar dan semua kran air yang ada dirumah tertutup maka pada saluran keluaran pompa akan timbul tekanan yang cukup besar, ketika tekanan yang dihasilkan melebihi tekan set yang ada pada sensor atau *Pressure Switch* maka sensor akan bekerja dan pompa air akan mati seketika, pompa air akan hidup lagi jika ada salah satu kran air terbuka disebabkan tekanan air sudah turun dan begitulah seterusnya.

Dengan demikian saat kita lupa untuk mematikan pompa air, maka mesin pompa air tidak akan terbakar disebabkan kerja yang terus menerus, dan kita tidak perlu memasang atau mencabut steker dari mesin pompa air sebab segalanya akan bekerja secara otomatis. Rangkain pompa air otomatis dapat dilihat pada Gambar 2.2.



**Gambar 2.2** Rangkaian Listrik Yang Terdapat Pada Sebuah Pompa Air

Keterangan Gambar 2.2 :

- a. AC 220 adalah tegangan sumber yang biasa digunakan dirumah-rumah.
- b. *Protector*, berfungsi sebagai pelindung motor agar tidak terbakar, jika terjadi panas yang berlebih pada gulungan motor, akibat dari pembebanan yang berlebihan, seperti bearing macet dll.
- c. *Capasitor*, berfungsi sebagai starting pada saat motor/pompa dihidupkan.
- d. *Main coil*, gulungan utama, berfungsi untuk membangkitkan putaran motor.
- e. *Sub coil*, gulungan bantu, berfungsi membantu membangkitkan putaran motor pada saat awal motor/pompa dihidupkan.

### 2.3 Sensor Ultrasonik HY-SRF05 [3]

HY-SRF05 merupakan sensor pengukur jarak yang menggunakan ultrasonik. Dimana prinsip kerja sensor Ultrasonik ini adalah pemancar (*transmitter*) mengirimkan seberkas gelombang ultrasonik, lalu diukur waktu yang dibutuhkan hingga datangnya pantulan dari obyek. Lamanya waktu ini sebanding dengan dua kali jarak sensor dengan obyek, sehingga didapat jarak sensor dengan obyek yang bisa ditentukan dengan persamaan. Sensor Ultrasonik HY-SRF05 dengan spesifikasi sebagai berikut:

- a. Bekerja pada tegangan DC 5 volt
- b. Beban arus lebih dari 2 mA
- c. Menghasilkan gelombang dengan frekuensi 40 KHz
- d. Jangkauan jarak yang dapat dideteksi 2 cm – 450 cm

Bentuk fisik Sensor Ultrasonik HY-SRF05 dapat dilihat pada Gambar 2.3.



**Gambar 2.3** Ultrasonik HY-SRF05

#### **2.4 Liquid Water Level Sensor Right Angle Float Switch [4]**

*Water Level Sensor* adalah alat yang digunakan untuk memberikan signal kepada alarm / *automation* panel bahwa permukaan air telah mencapai level tertentu. Sensor akan memberikan *signal dry contact* (NO/NC) ke panel. Cara kerja sensor ini adalah pada saat ketinggian air naik, maka secara otomatis bandul bermagnet akan ikut terangkat juga, dan ketika magnet berada pada level sensor berikutnya maka sensor tersebut akan aktif dan menyalakan lampu atau peralatan lainnya. Spesifikasi *water level sensor* ialah sebagai berikut :

- a. Tidak memerlukan catu daya untuk mendeteksi ketinggian air
- b. Terintegrasi dengan *internal switch*
- c. Tegangan pada kontak *switch* : 100VAC/VDC (*switching*) (maks)
- d. Arus pada kontak *switch* : 0,5A (*switching*)
- e. Bahan pelindung *switch* terbuat dari plastic
- f. Dapat dikonfigurasi *Normally Open* atau *Normally Close*

#### **2.5 Arduino Mega [6]**

Arduino mega 2560 adalah papan mikrokontroler ATmega2560 berdasarkan (*datasheet*) memiliki 54 digital pin *input / output* (dimana 15 dapat digunakan sebagai *output* PWM), 16 analog *input*, 4 UART (*hardware port serial*), osilator kristal 16 MHz, koneksi USB, jack listrik, *header* ICSP, dan tombol *reset*. Ini berisi semua yang diperlukan untuk mendukung mikrokontroler, hanya menghubungkannya ke komputer dengan kabel USB atau *power* dengan adaptor AC-DC atau baterai. Arduino Mega kompatibel dengan sebagian besar *shield*, dirancang untuk Arduino *Duemilanove* atau *Diecimila*. Bentuk fisik Arduino mega dapat dilihat pada Gambar 2.4 berikut.



**Gambar 2.4** Arduino Mega

Arduino Mega2560 berbeda dari semua *board* sebelumnya, tidak menggunakan *chip driver* FTDI USB-to-serial. Sebaliknya, fitur ATmega16U2 (ATmega8U2 dalam revisi 1 dan revisi 2 papan) diprogram sebagai konverter USB-to-serial. Revisi 2 dewan Mega2560 memiliki resistor menarik garis 8U2 HWB ke tanah, sehingga lebih mudah untuk dimasukkan ke dalam mode DFU.

Secara ringkas spesifikasi Arduino Mega ialah sebagai berikut :

- a. Mikrokontroler : Atmega2560
- b. Tegangan Operasi : 5V
- c. Tegangan Input : 7-12V
- d. Tegangan Input : 6-20V
- e. Jumlah Pin I/O Digital : 54 (15 diantaranya *output* PWM)
- f. Jumlah Pin Analog *Input*: 16
- g. Arus DC per Pin I/O : 40 mA
- h. Arus DC pada Pin 3.3V : 50 mA
- i. *Flash Memory* : 356KB(0.5 KB sebagai *bootloader*)
- j. SRAM : 8 KB
- k. EEPROM : 4 KB
- l. Clock Speed : 16 MHz

## 2.6 Arduino Shield Ethernet [5]

Arduino *shield ethernet* adalah modul yang dapat dipasangkan langsung di atas papan Arduino untuk menambahkan fungsi LAN / *ethernet* dalam proyek rangkaian elektronika yang menggunakan papan pengembangan Arduino. Dengan menggunakan modul komunikasi ini Arduino dapat dihubungkan dengan perangkat lain yang mendukung protokol TCP/IP atau UDP. Mendukung sampai dengan 4 koneksi secara bersamaan. *Shield ethernet* ini terhubung dengan *board* Arduino menggunakan header yang dapat ditumpuk (*stackable header*). Dengan



*header* ini *layout* pin akan tetap terjaga dan memungkinkan untuk *shield* lain ditumpukkan di atasnya. Contoh board *ethernet* ini ditunjukkan pada Gambar 2.5.



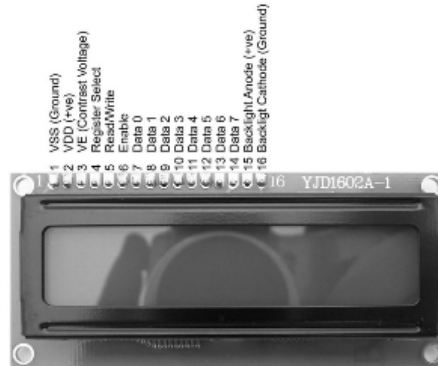
**Gambar 2.5** *Shield Ethernet*

Board *ethernet* ini mempunyai slot memori *micro-sd*, yang dapat dipergunakan untuk menyimpan file untuk digunakan di dalam jaringan. Cocok (*compatible*) digunakan dengan board Arduino Mega. *Duemilanove*, dan Mega (dengan *ethernet library*). Juga menyediakan tempat untuk modul *power-over-Ethernet* (poE) yang terpisah dan dapat disolder pada board ini untuk menyediakan daya dari kabel *ethernet* CAT 5. Sesuai dengan IEEE802.3af dan dapat digunakan dengan semua POE injectors yang tersedia saat ini..

## 2.7 LCD [2]

*LCD (Liquid Crystal Display)* berfungsi menampilkan suatu nilai hasil sensor, menampilkan teks, atau menampilkan menu pada aplikasi mikrokontroler. Sumber cahaya di dalam sebuah perangkat *LCD* adalah lampu neon berwarna putih dibagian belakang susunan kristal cair tadi. Titik cahaya yang jumlahnya puluhan ribu bahkan jutaan inilah yang membentuk tampilan. Kutub kristal cair yang dilewati arus listrik akan berubah karena pengaruh polarisasi medan magnetik yang timbul dan oleh karenanya akan hanya beberapa warna diteruskan sedangkan warna lainnya disaring.

Dalam Tugas Akhir ini menggunakan LCD 16x2 I2C. LCD ini membutuhkan *driver* supaya bisa dikoneksikan dengan sistem minimum dalam suatu mikrokontroler. *Driver* yang disebutkan berisi rangkaian pengaman, pengatur tingkat kecerahan maupun data, serta untuk mempermudah pemasangan di mikrokontroler. Konfigurasi pin LCD dapat dilihat pada Gambar 2.6.



**Gambar 2.6** Diagram Pin LCD

## 2.8 Modul I2C LCD Backpack [9]

Modul I2C LCD *backpack* adalah modul yang digunakan untuk mengurangi jumlah pin yang digunakan pada koneksi antara Arduino (atau mikrokontroler lainnya) dengan *character* LCD. Modul I2C LCD *backpack* yang digunakan dapat dilihat pada Gambar 2.7.



**Gambar 2.7** Modul I2C LCD Backpack

Untuk menggunakan modul ini, sebelumnya kita perlu menambahkan fungsi *library New Liquid Crystal* dengan cara mengunduh dan ekstrak *library* tersebut pada sub direktori dimana *library* arduino berada. Setelah itu kita perlu mengganti nama direktori *library Liquid Crystal* yang lama agar tidak bentrok dengan *library* yang baru. Serta kita tidak membutuhkan sub direktori `__MACOSX`, sehingga kita dapat menghapusnya. Selanjutnya, hubungkan arduino dengan I2C LCD *backpack*. Untuk Arduino Mega 2560, dimana pada arduino pin I2C sudah tersedia yaitu memiliki pada bagian ujung *digital* pin sehingga mudah dikenali (Pin dengan tulisan SDA dan SCL). Sedangkan untuk Arduino Mega, pin I2C berada pada pin analog SDA 20 dan SCL 21. Untuk memudahkan, berikut ini adalah Tabel 2.1 yang merupakan tabel koneksi antara I2C LCD backpack:

**Tabel 2.1** Konfigurasi I2C LCD *Backpack* pada Arduino

I2C LCD Backpack	Arduino Mega
GND	GND
VCC	5V
SDA	SDA 20
SCL	SCL 21

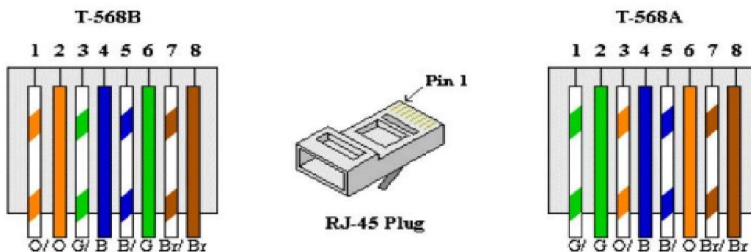
## 2.9 Komunikasi Serial Kabel Ethernet (RJ45) [7]

Untuk menghubungkan jaringan diperlukan kabel *Ethernet* yaitu kabel yang digunakan disebut kabel UTP (*Unshielded Twisted Pair*) dengan menggunakan konektor RJ45.

Kabel UTP mempunyai delapan pin (4 pasang).

- Pin1 dengan warna hijau-putih (TD+)
- Pin2 dengan warna hijau (TD-)
- Pin3 dengan warna orange-putih (RD+)
- Pin4 dengan warna biru (NC)
- Pin5 dengan warna biru-putih (NC)
- Pin6 dengan warna orange (RD-)
- Pin7 dengan warna coklat-putih (NC)
- Pin8 dengan warna coklat (NC)

Konfigurasi pin kabel UTP dapat dilihat pada Gambar 2.8.



**Gambar 2.8** Kabel UTP dengan standar T-568B dan T-568B serta konektor RJ 45

Ada tiga cara pemasangan kabel UTP:

### 1. *Straigh Through*

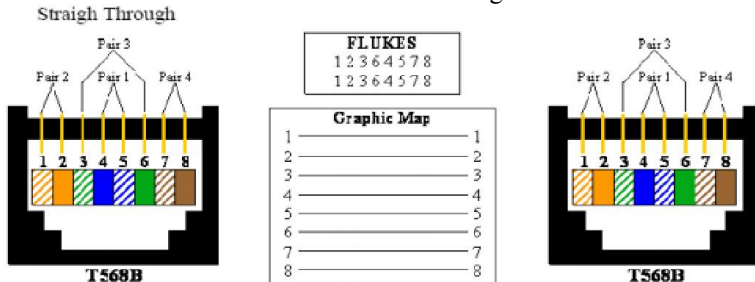
Pengkabelan jenis ini biasanya diperuntukkan untuk menghubungkan peralatan yang berbeda jenis. Misal untuk menghubungkan PC dengan *hub*, *switch* dan *router*, *switch* dan PC dan sebagainya.

### 2. *Cross Over*

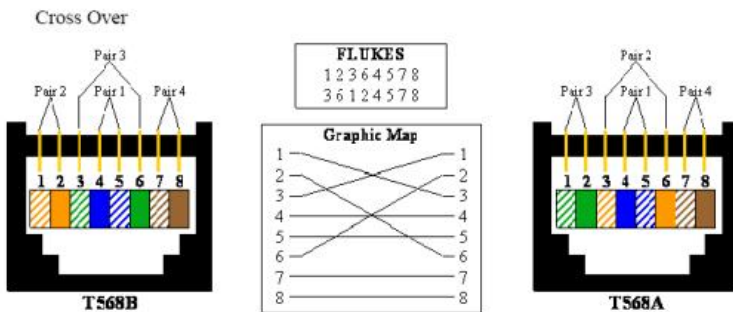
Pengkabelan jenis ini biasanya digunakan untuk menghubungkan peralatan sejenis. Misal untuk menghubungkan PC dengan PC, *hub* dengan *hub* dan sebagainya.

### 3. *Rollover*

Pengkabelan jenis ini merupakan pengkabelan khusus. Misalnya untuk menghubungkan antar *switch*. Skema dari jenis kabel di atas dapat dilihat Gambar 2.9 untuk *straight through* dan Gambar 2.10 untuk *cross over* adalah sebagai berikut :



**Gambar 2.9** Kabel untuk *Straigh Through*



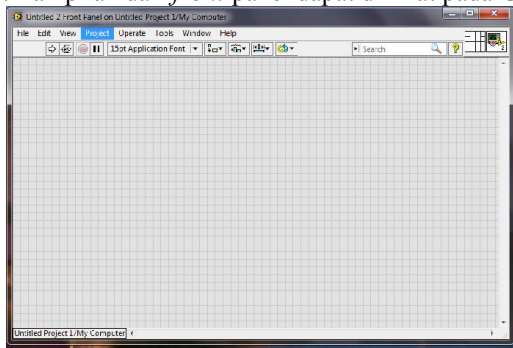
**Gambar 2.10** Kabel untuk *Cross Over*

## 2.10 LabVIEW [8]

*LabVIEW* adalah sebuah *software* pemrograman yang diproduksi oleh *National instruments* dengan konsep yang berbeda. Seperti bahasa pemrograman lainnya yaitu C++, *Matlab* atau *Visual basic*, *LabVIEW* juga mempunyai fungsi dan peranan yang sama, perbedaannya bahwa *LabVIEW* menggunakan bahasa pemrograman berbasis grafis atau blok diagram sementara bahasa pemrograman lainnya menggunakan basis *text*. Program *LabVIEW* dikenal dengan sebutan Vi atau *Virtual instruments* karena penampilannya dapat meniru sebuah *instrument*. Pada *LabVIEW*, *user* pertama-tama membuat *user interface* atau *front panel* dengan menggunakan *control* dan indikator, yang dimaksud dengan *control* adalah *knobs*, *push buttons*, *dials* dan peralatan input lainnya sedangkan yang dimaksud dengan indikator adalah *graphs*, LEDs dan peralatan *display* lainnya. Setelah menyusun *user interface*, lalu *user* menyusun blok diagram yang berisi kode-kode VIs untuk mengontrol *front panel*. *Software LabVIEW* terdiri dari tiga komponen utama, yaitu :

### 1. Front panel

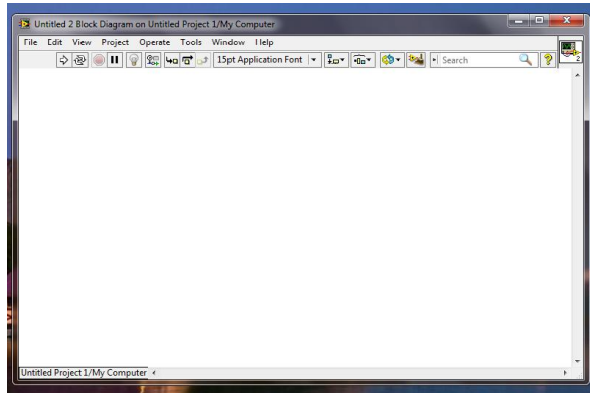
*Front panel* adalah bagian *window* yang berlatar belakang abu-abu serta mengandung *control* dan indikator. *Front panel* digunakan untuk membangun sebuah VI, menjalankan program dan mendebug program. Tampilan dari *front panel* dapat dilihat pada Gambar 2.11



Gambar 2.11 Front Panel *LabVIEW*

### 2. Blok Diagram

Blok diagram adalah bagian *window* yang berlatar belakang putih berisi *source code* yang dibuat dan berfungsi sebagai instruksi untuk *front panel*. Tampilan dari blok diagram dapat dilihat pada Gambar 2.12



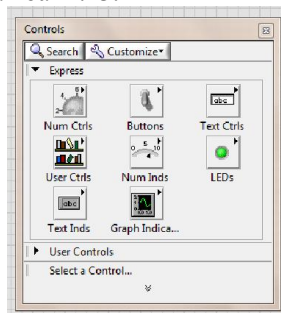
**Gambar 2.12** Blok Diagram *LabVIEW*

### 3. ***Control dan Functions Palette***

*Control dan Functions Palette* digunakan untuk membangun sebuah Vi. Berikut ini adalah pengertian dari *Control dan Functions Palette*.

#### a. *Control Palette*

*Control Palette* merupakan tempat beberapa *control* dan indikator pada *front panel*, *control palette* hanya tersedia di *front panel*, untuk menampilkan *control palette* dapat dilakukan dengan mengklik *view >> control palette* pada *front panel* atau klik kanan pada *front panel*. Tampilan *control palette* dapat dilihat pada Gambar 2.13.

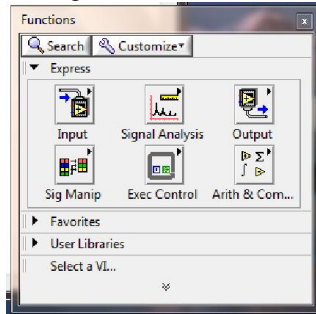


**Gambar 2.13** *Control Palette* Pada *Lab Front Panel LabVIEW*

#### b. *Functions Palette*

*Functions Palette* di gunakan untuk membangun sebuah blok diagram, *functions palette* hanya tersedia pada blok diagram,

untuk menampilkannya dapat dilakukan dengan mengklik *view* >> *functions palette* pada lembar kerja blok diagram atau klik kanan pada lembar kerja blok diagram. Tampilan *functions palette* dapat dilihat pada Gambar 2.14.



**Gambar 2.14** Tampilan *Functions Palette LabVIEW*

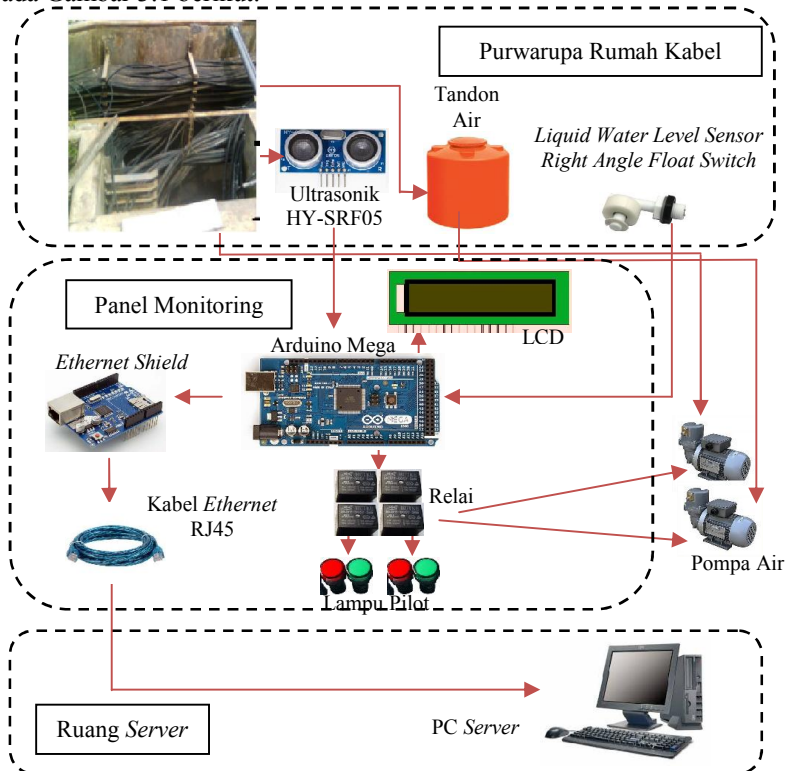
## BAB III

### PERANCANGAN DAN PEMBUATAN ALAT

Dari teori penunjang yang didapat, selanjutnya dilakukan langkah perancangan dan pembuatan alat yang dibutuhkan pada Tugas Akhir ini. Perancangan yang dilakukan terdiri atas empat bagian, yaitu perancangan sistem keseluruhan, perancangan mekanik, perancangan *hardware* meliputi perancangan rangkaian elektronika dan perancangan *software* meliputi perancangan program dan *interface* yang digunakan.

#### 3.1 Perancangan Keseluruhan

Secara umum rancangan alat pada tugas akhir ini dapat dilihat pada Gambar 3.1 berikut:



**Gambar 3.1** Diagram Fungsional Sistem Alat Secara Keseluruhan



Gambar 3.1 menunjukkan keseluruhan sistem dari alat purwarupa monitoring dan kontrol pompa rumah kabel bawah tanah . Dari gambar tersebut dapat dilihat bahwa terdapat 3 bagian skematik utama yang terdiri dari :

1. Purwarupa rumah kabel : dapat menggambarkan rumah kabel yang akan dimonitoring kinerjanya yaitu ketinggian air yang baik oleh pompa air maupun peralatan-peralatan yang lain. Terdapat tandon yang dibentuk sedemikian rupa agar dapat menggambarkan aliran air dari rumah kabel menuju tandon. Dan tentunya dilengkapi dengan komponen yang menggambarkan peralatan-peralatan yang ada di rumah kabel.
2. Panel monitoring : dapat dilihat bahwa ketika Ultrasonik mengukur ketinggian yang melebihi batas yang diijinkan maka secara otomatis Ultrasonik mentransmisikan data ke Arduino. Data yang diterima oleh Arduino akan diolah, kemudian Arduino akan mengirimkan sinyal ke *driver* relai untuk mengaktifkan relai yang akan mengaktifkan lampu pilot dan pompa air. Disaat yang bersamaan akan dikirim melalui kabel *ethernet* yang terhubung dengan *ethernet shield*.
3. Ruang *server* : pada ruang server akan menerima data yang dikirim melalui kabel *ethenet* yang terhubung dengan *ethernet shield*. Data akan diterima langsung yang akan ditampilkan pada PC *server* melalui *interface LabVIEW*.

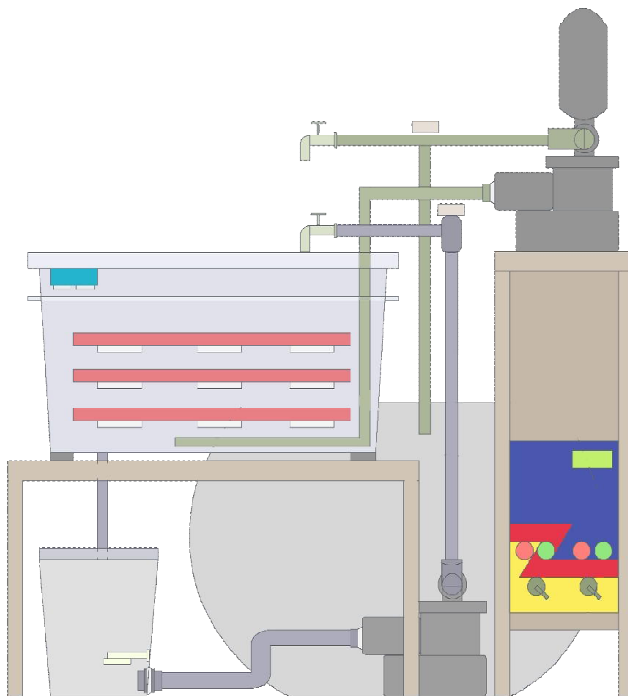
### 3.2 Perancangan Mekanik

Perancangan Mekanik berupa purwarupa rumah kabel bawah tanah beserta beberapa komponen yang mewakili peralatan-peralatan utama yang ada. Purwarupa rumah kabel ini dibuat dari bahan box plastik berukuran 60L pada bagian atas dan tandon dibuat dari bahan ember plastik tertutup berukuran 7,5 L pada bagian bawah. Tujuan pembuatan purwarupa ini adalah untuk menggambarkan aliran air pada rumah kabel dan skenario pengaturan tinggi permukaan air di rumah kabel. Dibuat lubang antar rumah kabel dan tandon untuk menggambarkan air yang mengalir pada rumah kabel menuju tandon.

Purwarupa rumah kabel dilengkapi dengan beberapa komponen-komponen yang menggambarkan peralatan di rumah kabel. Komponen-komponen inilah yang akan dimonitoring kinerjanya. Untuk mempermudah skenario pengaturan air menggunakan sensor Ultrasonik mengukur ketinggian yang melebihi batas yang diijinkan maka secara otomatis Ultrasonik mentransmisikan data ke Arduino. Data yang diterima oleh Arduino akan diolah, kemudian Arduino akan

mengirimkan sinyal ke *driver* relai untuk mengaktifkan relai yang akan mengaktifkan lampu pilot dan pompa air. Terdapat 2 buah pompa yang diletakkan pada rumah kabel dan tandon. Pompa pertama pada tandon digunakan untuk memindahkan air kembali pada rumah kabel, sedangkan pompa kedua pada rumah kabel digunakan untuk memindahkan air dari satu bagian ke bagian lain. Dibuat skenario air yang dipindahkan dari tandon dialirkan kembali ke rumah kabel adalah kita tidak perlu mengisi air pada rumah kabel.

Secara umum purwarupa rumah kabel pada tugas akhir ini dapat dilihat pada Gambar 3.2 berikut:



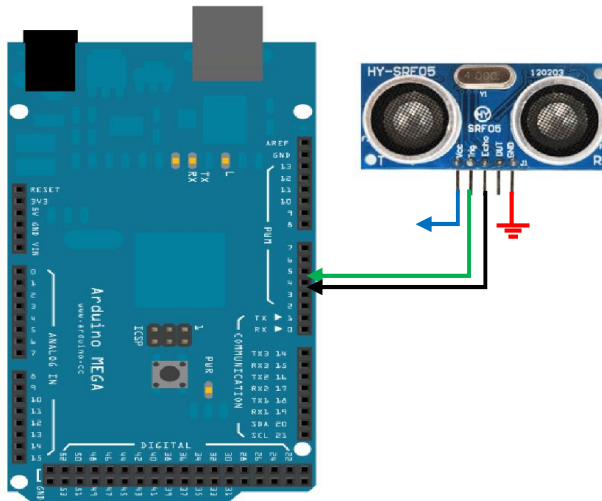
**Gambar 3.2** Purwarupa Rumah Kabel

### 3.3 Perancangan *Hardware*

Perancangan *Hardware* yang dibahas terdiri dari perancangan rangkaian sensor ketinggian air, perancangan LCD, perancangan modul Arduino Mega, dan perancangan modul relai.

#### 3.3.1 Perancangan Sensor Ultrasonik HY-SRF05

Pada rumah kabel sensor ketinggian air yang dipakai adalah Ultrasonik HY-SRF05. Dimana prinsip kerja sensor Ultrasonik ini adalah pemancar (*transmitter*) mengirimkan seberkas gelombang ultrasonik, lalu diukur waktu yang dibutuhkan hingga datangnya pantulan dari obyek. Ultrasonik HY-SRF05 untuk jangkauan jarak yang dapat dideteksi 2 cm – 450 cm. Sensor ultrasonik memiliki 5 pin dimana pin 1 merupakan VCC, pin 2 adalah pin *trig*, pin 3 adalah *echo*, pin 4 adalah pin *OUT*, dan pin 5 adalah GND. Konfigurasi pin dapat dilihat pada Gambar 3.3



**Gambar 3.3** Konfigurasi Pin Sensor Ultrasonik HY-SRF05

Keterangan Gambar 3.3 :

Arduino Mega 2560

■ VCC 5 Volt

■ Pin 5

■ Pin 4

■ Ground

HY-SRF05

■ VCC

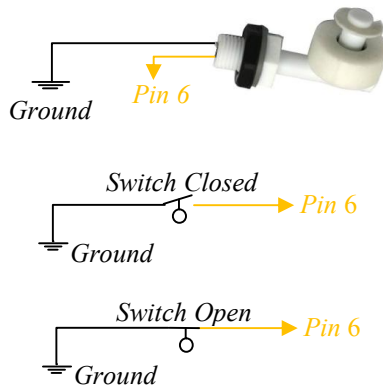
■ Trig

■ Echo

■ Ground

### 3.3.2 Perancangan *Liquid Water Level Sensor Right Angle Float Switch*

Pada tandon sensor yang dipakai adalah *liquid water level sensor right angle float switch*. Dimana digunakan untuk memberikan signal kepada alarm / *automation* panel bahwa permukaan air telah mencapai level tertentu. Sensor akan memberikan *signal dry contact* (NO/NC) ke panel. Cara kerja sensor ini adalah pada saat ketinggian air naik, maka secara otomatis bandul bermagnet akan ikut terangkat juga, dan ketika magnet berada pada level sensor berikutnya maka sensor tersebut akan aktif. Perancangan sensor ini dapat dilihat pada Gambar 3.4

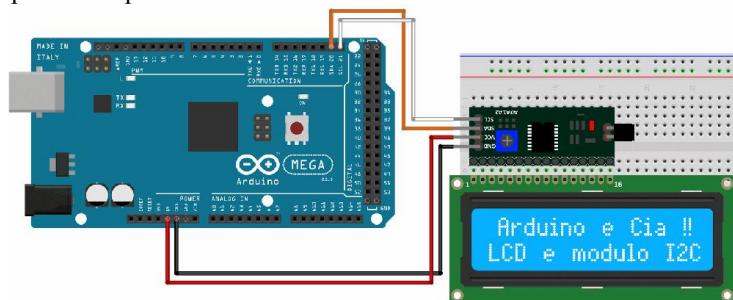


**Gambar 3.4** Konfigurasi Pin *Liquid Water Level Sensor Right Angle Float Switch*

### 3.3.3 Perancangan LCD

LCD (*Liquid Crystal Display* atau dapat di bahasa Indonesia-kan sebagai tampilan Kristal Cair) adalah suatu jenis media tampilan yang menggunakan Kristal Cair sebagai penampil utama. LCD bisa memunculkan gambar atau tulisan dikarenakan terdapat banyak sekali titik cahaya (piksel) yang terdiri dari satu buah Kristal cair sebagai sebuah titik cahaya. LCD yang digunakan pada Tugas Akhir ini adalah i2C LCD dengan banyak karakter 16x2. Maksudnya adalah 16 kolom dan 2 baris. Tegangan yang diberikan pada LCD tersebut adalah sebesar 5 Volt DC. Adapun I2C LCD *backpack* ini bertujuan untuk mengurangi jumlah pin yang digunakan pada koneksi antara Arduino (atau

*microcontroller* lainnya) dengan *character* LCD. Konfigurasi pin LCD dapat dilihat pada Gambar 3.5



**Gambar 3.5** Konfigurasi Pin LCD

Keterangan Gambar 3.5 :

Arduino Mega 2560

■ VCC 5 Volt

□ SCL 21

■ SDA 20

■ Ground

I2C LCD

■ VCC

□ SCL

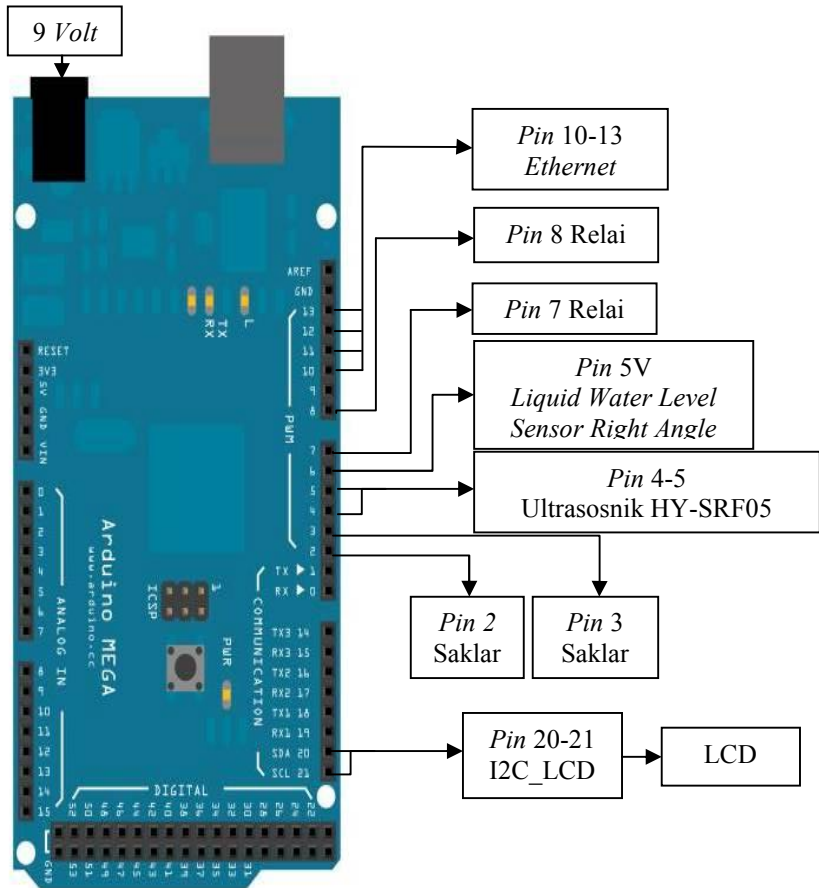
■ SDA

■ Ground

### 3.3.4 Perancangan Modul Arduino Mega

Dalam perancangan perangkat keras terdapat rangkaian Arduino sistem minimum Atmega2560. Sistem minimum Atmega2560 berfungsi untuk menerima data yang dikirimkan oleh sensor ketinggian air. Data yang diterima akan diproses dan ditransmisikan ke PC *server* melalui komunikasi kabel *ethernet* yang terhubung dengan *ethernet shield*.

Pada Arduino Mega ini bisa menggunakan tegangan input dari 5V hingga 12V. Untuk penggunaan kaki-kaki pada Arduino Mega, pada Pin tegangan operasi 5V digunakan untuk mengoperasikan sensor air yang dihubungkan dengan relai. Pin 8 sebagai keluaran untuk menyalakan relai untuk mengaktifkan lampu pilot dan pompa air. Pin 6 dan 7 sebagai keluaran untuk menyalakan sensor ultrasonik apabila syarat program terpenuhi. *Pin* SDA dan SCL sebagai masukan I2C LCD yang dihubungkan dengan LCD. Sedangkan Pin 10,11,12, dan 13 digunakan untuk komunikasi melalui *shield ethernet*. Penggunaan Pin pada Arduino dapat dilihat pada Gambar 3.6.

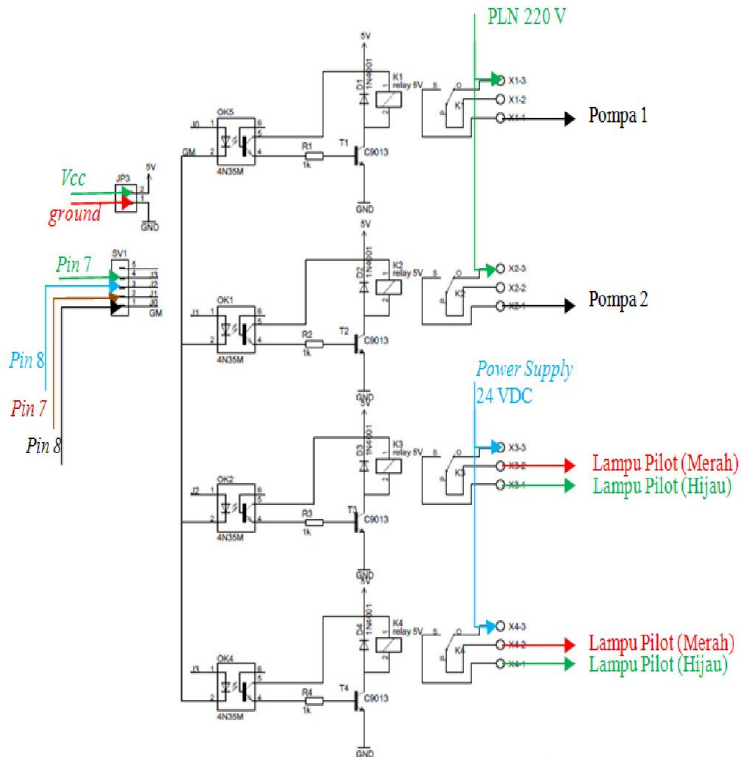


**Gambar 3.6** Penggunaan Pin Pada Arduino Mega

### 3.3.6 Perancangan Modul Relai

Relai berfungsi sebagai modul penghubung dan pemisah rangkaian. Untuk menggerakkan relai maka diperlukan *driver* relai. *Driver* relai akan menerima sinyal dari Arduino. Modul relai pada alat ini digunakan untuk menyalakan lampu pilot dan pompa air. Untuk membuat *driver* relai digunakan resistor, dioda, dan transistor jenis NPN. Dioda sangat penting untuk menjaga adanya arus balik ke Arduino

ketika *switching* terjadi. Untuk transistor NPN ini berfungsi sebagai saklar elektronik yang akan mengalirkan arus jika terdapat arus bias pada kaki basisnya, dan akan menyumbat arus jika tidak terdapat arus bias pada kaki basisnya. Gambar untuk *driver* relai seperti Gambar 3.7.



**Gambar 3.7** *Driver Relai*

### 3.4 Perancangan *Software*

Perancangan perangkat lunak ini menggunakan *software* Arduino IDE untuk memprogram Arduino dan *software* LabVIEW yang digunakan untuk menampilkan hasil pembacaan ketinggian air yang didapatkan oleh purwarupa rumah kabel.

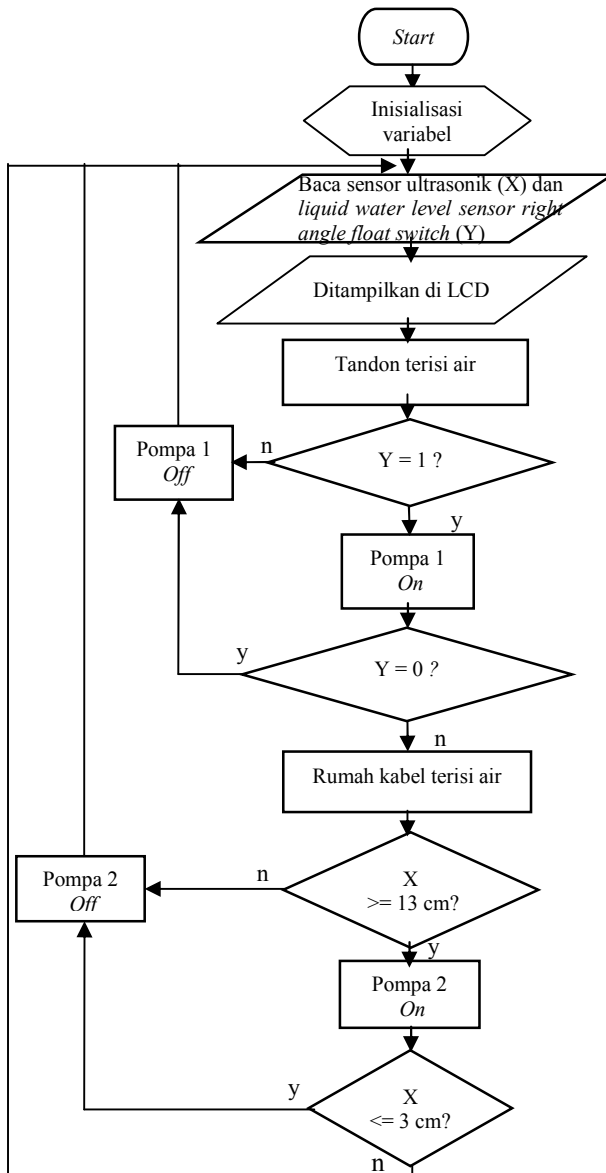
### 3.4.1 Program pada Arduino Mega

Editor teks ini mempunyai fasilitas untuk *cut/paste* dan *search/replace*. Area pesan berisi umpan balik ketika menyimpan dan mengunggah file, dan juga menunjukkan jika terjadi error. Program pada Arduino Mega menggunakan program Arduino IDE dengan bahasa pemrograman bahasa C. Arduino IDE adalah sebuah editor yang digunakan untuk menulis program, meng-*compile*, dan mengunggah ke papan Arduino. Arduino IDE terdiri dari editor teks untuk menulis kode, area pesan, *console teks*, *toolbar* dengan tombol-tombol untuk fungsi umum, dan sederetan menu. *Software* yang ditulis menggunakan Arduino dinamakan *sketches*. *Sketches* ini ditulis di editor teks dan disimpan dengan file yang berekstensi .ino. *Flow Chart* pemrogramannya seperti terlihat pada Gambar 3.8.

Dari diagram alir Gambar 3.8, algoritma dari perancangan perangkat lunak untuk Arduino ini adalah :

1. Arduino aktif melakukan inisialisasi variabel dengan permisalan :  
X = Sensor HY-SRF05  
Y = *Liquid Water Level Sensor Right Angle Float Switch*
2. Baca sensor pada kedua sensor yang akan ditampilkan di LCD.
3. Tandon terisi air.
4. Sensor Y bekerja jika sensor Y *switch open* maka pompa 1 menyala, apabila sensor Y *switch close* maka pompa 1 mati.
5. Rumah kabel terisi air.
6. Sensor X bekerja jika hasil ketinggian air dari sensor X lebih besar sama dengan 13 cm maka pompa 2 menyala, apabila hasil ketinggian air dari sensor X kurang dari sama dengan 3 cm maka pompa 2 mati.





**Gambar 3.8** *Flow Chart* Arduino

### 3.4.2 LabVIEW

*LabVIEW* adalah sebuah *software* pemrograman yang diproduksi oleh *National instruments* dengan konsep yang berbeda. *LabVIEW* juga mempunyai fungsi dan peranan yang sama, perbedaannya bahwa *LabVIEW* menggunakan bahasa pemrograman berbasis grafis atau blok diagram sementara bahasa pemrograman lainnya menggunakan basis *text*.

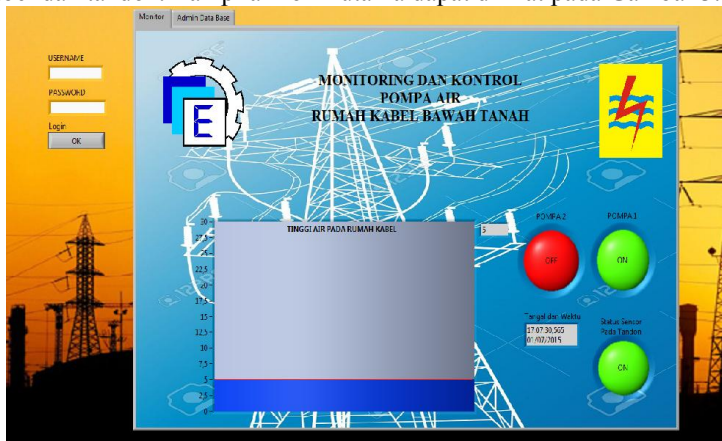
Pada *LabVIEW*, user pertama-tama membuat *user interface* atau *front panel* dengan menggunakan *control* dan indikator, yang dimaksud dengan *control* adalah *knobs*, *push buttons*, *dials* dan peralatan input lainnya sedangkan yang dimaksud dengan indikator adalah *graphs*, LEDs dan peralatan *display* lainnya. Setelah menyusun *user interface*, lalu user menyusun blok diagram yang berisi kode-kode VIs untuk mengontrol *front panel*

### 3.4.3 Tampilan Monitoring

Tampilan monitoring yang telah dibuat menggunakan *software LabVIEW* form utama yang di tampilkan.

#### a. Form utama

Form utama ini berfungsi untuk menampilkan data waktu, tanggal, keterangan pompa yang bekerja, nilai ketinggian air pada rumah kabel dan tandon. Tampilan form utama dapat dilihat pada Gambar 3.9.

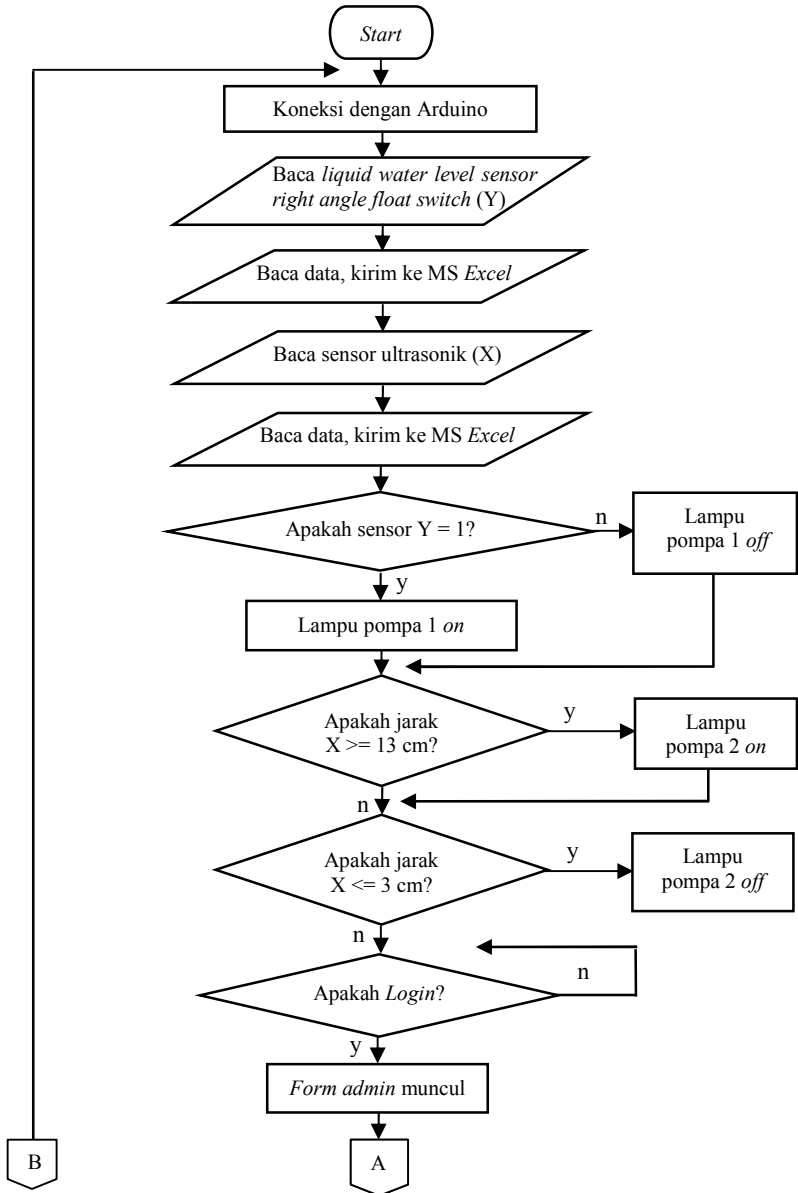


**Gambar 3.9** Tampilan Form Utama

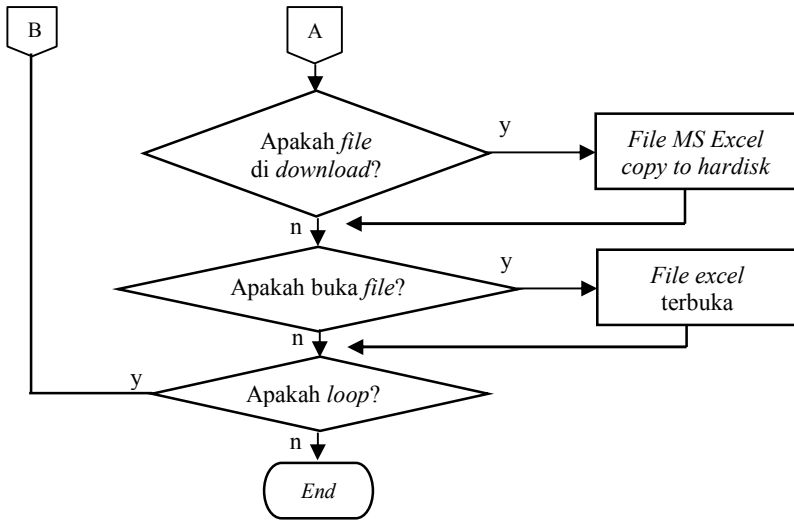
Sedangkan algoritma untuk program tampilan *software LabVIEW* adalah sebagai berikut :

1. *LabVIEW* melakukan koneksi dengan Arduino.
2. Baca sensor X dilihat hasil datanya dan dikirim ke *Microsoft Excel* dan menampilkan data *delay* 500 ms.
3. Baca sensor Y dilihat hasil datanya dan dikirim ke *Microsoft Excel* dan menampilkan data *delay* 500 ms.
4. Sensor Y bekerja jika sensor Y bernilai 1 maka lampu pompa 1 menyala, apabila sensor Y bernilai 0 maka lampu pompa 1 mati.
5. Sensor X bekerja jika hasil ketinggian air dari sensor X lebih besar sama dengan 13 cm maka lampu pompa 2 menyala, apabila hasil ketinggian air dari sensor X kurang dari sama dengan 3 cm maka lampu pompa 2 mati.
6. Untuk mendapatkan *database* harus *login* terlebih dahulu.
7. Setelah *login*, bisa membuka *admin database* dilanjutkan klik *download*.
8. Pilih *file database* terbaru dan klik buka *file* untuk membuka.

*Flowchart* tampilan *software LabVIEW* dapat dilihat pada Gambar 3.10 dan Gambar 3.11.



**Gambar 3.10** Flow Chart LabVIEW



**Gambar 3.11** Lanjutan *Flow Chart LabVIEW*

## BAB IV PENGUJIAN DAN ANALISA DATA

Untuk mengetahui hasil dari pembuatan alat dalam kegiatan tugas akhir ini, maka perlu dilakukan pengujian serta menganalisa data yang diperoleh. Pada bab ini akan dibahas mengenai hasil pengujian serta analisa data yang diperoleh. Gambar 4.1 berikut ini menampilkan hasil pembuatan *hardware* secara keseluruhan alat monitoring dan kontrol pompa air pada rumah kabel bawah tanah.



**Gambar 4.1** Hasil Pembuatan *Hardware* Secara Keseluruhan

Pengujian merupakan salah satu langkah yang harus dilakukan untuk mengetahui apakah sistem yang telah dibuat sesuai dengan yang direncanakan. Kesesuaian sistem dengan perencanaan dapat dilihat dari hasil-hasil yang dicapai pada pengujian sistem. Pengujian juga bertujuan untuk mengetahui kelebihan dan kekurangan dari sistem yang telah dibuat. Hasil pengujian tersebut akan dianalisa untuk mengetahui penyebab terjadinya kekurangan atau kesalahan dalam sistem.

Pengujian tersebut meliputi:

1. Pengujian Arduino Mega
2. Pengujian Relai
3. Pengujian Sensor Ultrasonik HY-SRF05
4. Pengujian *Liquid Water Level Sensor Right Angle Float Switch*
5. Pengujian Komunikasi dengan *Ethernet*
6. Pengujian HMI
7. Pengujian Alat Keseluruhan

#### 4.1 Pengujian Arduino Mega

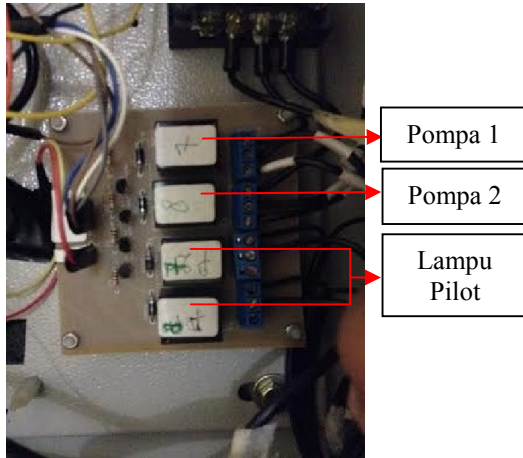
Untuk menguji arduino mega, maka perlu diisi dengan program. Sebuah program sederhana dibuat untuk melakukan penggunaan *port – port* arduino dengan cara member *logic* 0 dan 1. Untuk menandakan apakah arduino tersebut bekerja, maka digunakan LCD sebagai indikator dapat dilihat pada Gambar 4.2



**Gambar 4.2** Pengujian Arduino

#### 4.2 Pengujian Relai

Rangkaian *driver* relai dihubungkan dengan *led* sebagai indikator relai aktif atau tidak. Pengujian driver relai ini untuk membuktikan spesifikasi kerja dari *driver* relai sehingga dapat diketahui kondisi minimal dari relai agar relai dapat bekerja. *Driver* relai akan menerima sinyal dari arduino. Modul relai pada alat ini digunakan untuk menyalakan lampu pilot dan pompa air. Gambar pengujian rangkaian relai dapat dilihat pada Gambar 4.3. Hasil pengujian *driver* relai dapat dilihat pada Gambar 4.4.



**Gambar 4.3** Pengujian Rangkaian Relai



**Gambar 4.4** Hasil Pengujian Relai

Gambar pengujian diatas menjelaskan bahwa relai yang digunakan bekerja dengan baik dengan menggunakan lampu indikator sebagai perubahan status saklar relai

#### **4.3 Pengujian Sensor Ultrasonik HY-SRF05**

Sensor Ultrasonik HY-SRF05 merupakan alat yang digunakan untuk mengukur ketinggian yang ada. Dimana prinsip kerja sensor Ultrasonik ini adalah pemancar (*transmitter*) mengirimkan seberkas gelombang ultrasonik, lalu diukur waktu yang dibutuhkan hingga datangnya pantulan dari obyek. Lamanya waktu ini sebanding dengan



dua kali jarak sensor dengan obyek, sehingga didapat jarak sensor dengan obyek yang bias ditentukan dengan persamaan. Gelombang ultrasonik merupakan gelombang akustik yang memiliki frekuensi mulai 20 kHz hingga sekitar 20 MHz. Secara matematis besarnya jarak dapat dihitung sebagai berikut :

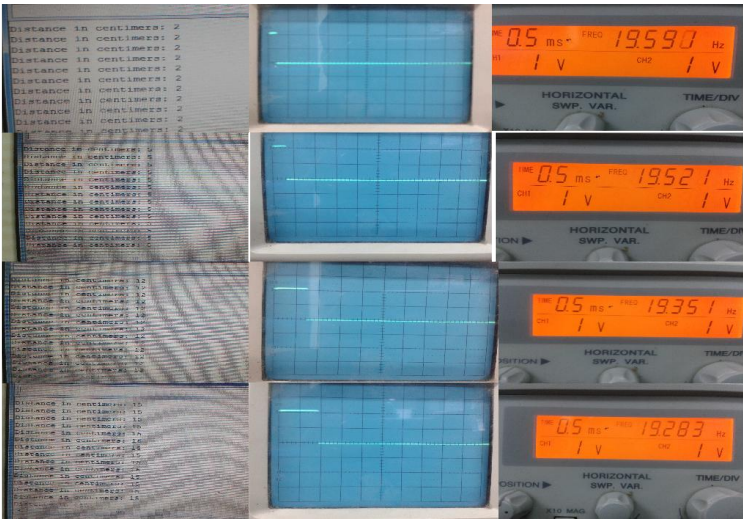
$$s = \frac{v.t}{2} \dots\dots\dots(4.1)$$

- dimana :  $s$  = Jarak dalam satuan meter  
(jarak antara sensor ultrasonik dengan bidang pantul)  
 $v$  = Kecepatan suara yaitu 344 m/detik  
 $t$  = Waktu tempuh dalam satuan detik  
(selisih antara pemancaran gelombang ultrasonik sampai diterima kembali oleh bagian penerima ultrasonik)

Untuk menghitung ketinggian air pada rumah kabel dapat dirumuskan sebagai berikut :

$$\text{Ketinggian air } (h_{\text{air}}) = 27 - s \dots\dots\dots(4.2)$$

- dimana :  $h_{\text{air}}$  = Tinggia air dalam satuan cm  
27 = Jarak dasar rumah kabel ke sensor Ultrasonik dalam satuan cm  
 $s$  = Jarak air yang terukur di rumah kabel

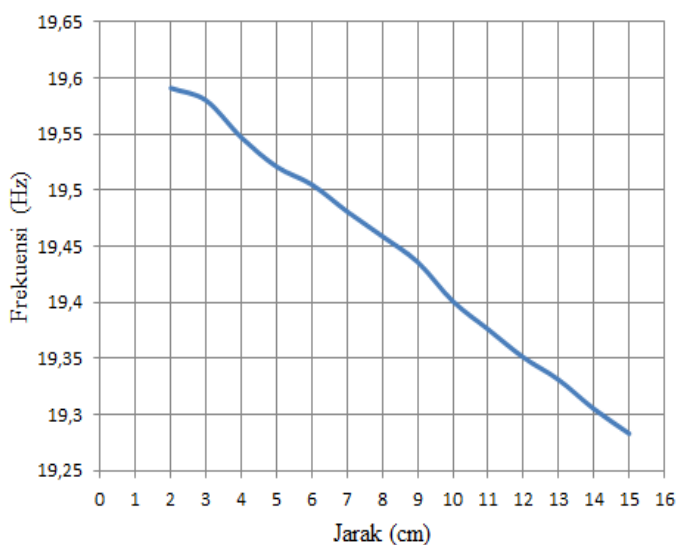


**Gambar 4.5** Pengujian Sensor Ultrasonik HY-SRF05

Dari hasil pengujian diatas pada Gambar 4.5 didapatkan jarak (dalam satuan centimeter) , lebar pulsa dan frekuensinya. Didapatkan beberapa pengukuran dalam jarak tertentu pada Tabel 4.1 dibawah ini :

**Tabel 4.1** Hasil Pengukuran Sensor Ultrasonik HY-SRF05

Jarak yang terukur (cm)	Frekuensi (Hz)
2	19,591
3	19,580
4	19,547
5	19,521
6	19,505
7	19,481
8	19,459
9	19,436
10	19,401
11	19,376
12	19,351
13	19,331
14	19,305
15	19,283

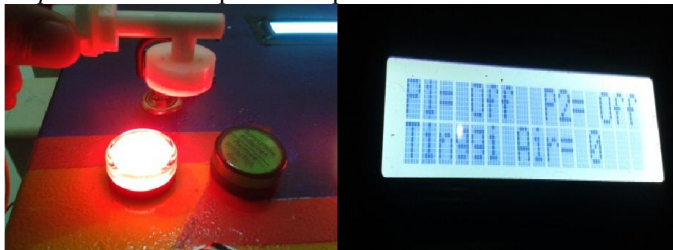


**Gambar 4.6** Kurva Pengujian Sensor HY-SRF05

Hasil pengujian sensor ultrasonik dapat dijelaskan bahwa apabila jarak yang diukur semakin kecil maka frekuensinya semakin besar, dan apabila jarak yang terukur semakin besar maka frekuensinya semakin kecil. Untuk kurva pengujian sensor ultrasonik HY-SRF05 dapat dilihat pada Gambar 4.6.

#### 4.4 Pengujian *Liquid Water Level Sensor Right Angle Float Switch*

Sensor ini menggunakan prinsip dimana digunakan untuk memberikan *signal* kepada alarm / *automation* panel bahwa permukaan air telah mencapai level tertentu. Pentingnya pengisian dan pengontrolan air diperlukan untuk mengefektifkan penggunaan air. Untuk pengontrolan air pada sistem ini menggunakan sensor untuk mendeteksi level ketinggian air pada tandon. Ketika sensor bergerak karena air akan bersifat *switch open* dan jika tidak bergerak karena air akan bersifat *switch closed* seperti sifat pada *datasheet*. *Switch open* dan *switch closed* yang merupakan keluaran dari sensor dibaca dan diolah oleh arduino yang selanjutnya memicu *driver* relai untuk menghidupkan atau mematikan pompa air. Untuk hasil pengujian pada saat sensor keadaan *switch closed* bernilai 0 dapat dilihat pada Gambar 4.7 dan keadaan *switch open* bernilai 1 dapat dilihat pada Gambar 4.8



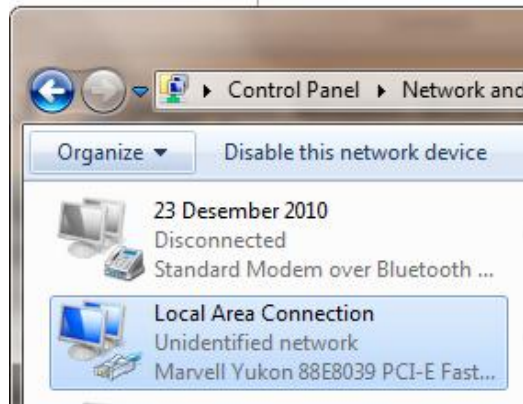
**Gambar 4.7** Keadaan *Switch Closed*



**Gambar 4.8** Keadaan *Switch Open*

#### 4.5 Pengujian Komunikasi dengan Kabel *Ethernet*

Untuk mengetahui apakah komunikasi kabel *ethernet* dapat digunakan perlu dilakukan suatu pengujian. Pengujian meliputi pengujian koneksi dan pengujian pengiriman data. Pengujian koneksi dilakukan untuk melihat apakah kabel *ethernet* dapat terhubung dengan computer server atau tidak. Pengujian koneksi dapat dilihat pada Gambar 4.9.

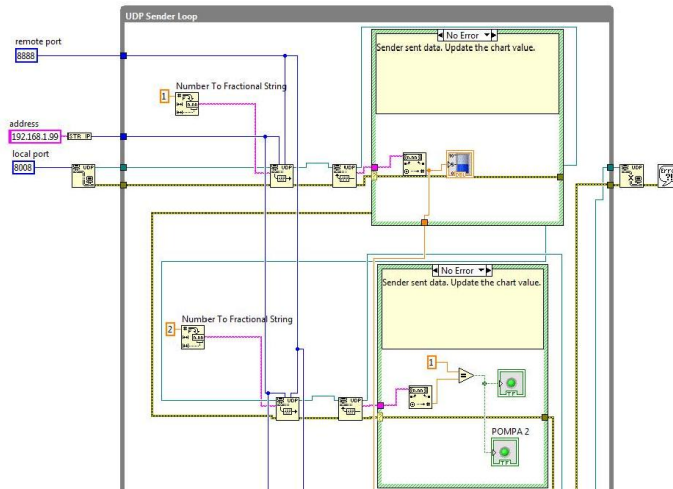


**Gambar 4.9** Pengujian Koneksi

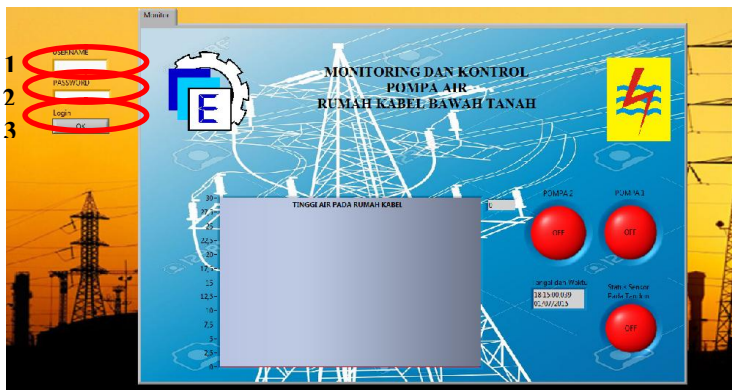
Pada kotak dialog seperti Gambar 4.9 seperti di atas, menggambarkan koneksi kabel *ethernet* telah berhasil terkoneksi. Setelah terkoneksi perlu diuji apakah kabel *ethernet* sudah benar benar terkoneksi atau belum. Uji koneksi dapat menggunakan pengujian komunikasi juga dapat dilakukan dengan cara klasik, yaitu dengan menggunakan perintah cmd. Pengujian ini disebut juga sebagai *PingTest*. *PingTest* dilakukan dengan cara berikut :

1. Membuka kotak dialog run dan menetikkan perintah cmd.
2. Setelah itu akan muncul kotak dialog dan kita ketikkan perintah *Ping 192.168.1.46* (IP yang sudah di-*setting* sebelumnya).
3. Apabila koneksi berhasil, maka akan memberikan respon reply. Hasil *PingTest* lewat cmd bisa dilihat pada Gambar 4.10.





**Gambar 4.11** Blok Diagram *host IP*



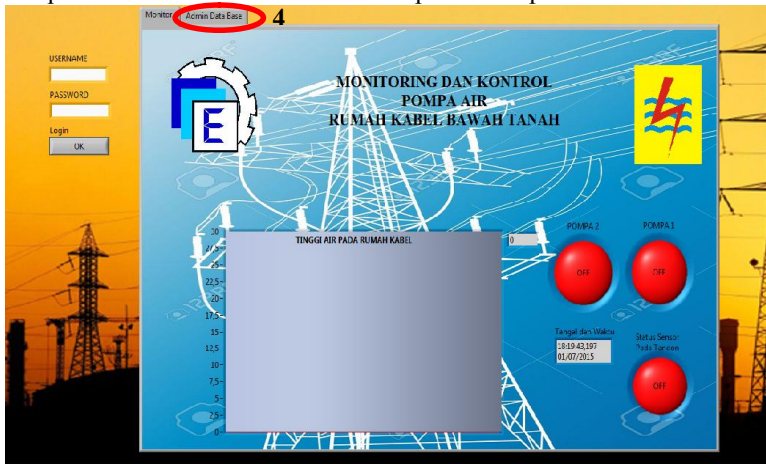
**Gambar 4.12** Tampilan Utama Monitoring

Keterangan Gambar 4.11 :

1. Mengisi *username* = sukolilo
2. Mengisi *password* = 112
3. Klik OK

#### 4.6.2 Pengujian Database

Seluruh proses akuisisi data akan masuk ke dalam sebuah *database*. *Database* ini dibagi menjadi dua bagian yaitu *database* untuk *admin* dan *monitor*. *Database* ini disimpan menggunakan *Ms. Excel*. Pengujian *database* dilakukan untuk memastikan bahwa penyimpanan data ke dalam *database* tersebut berjalan dengan baik. Untuk menyimpan seluruh proses tersebut maka yang pertama dilakukan adalah menekan tombol *Download* pada *Admin Database* pada HMI sehingga *database* dapat tersimpan dalam *Ms. Excel*. Tampilan HMI untuk *admin database* dapat dilihat pada Gambar 4.13. Tampilan HMI untuk *download database* dapat dilihat pada Gambar 4.14. Dalam *database* ini dapat dilihat rekaman hasil monitoring dengan melihat tanggal dan waktu pengambilan data pengukuran ketinggian air. Tampilan database dalam *Ms. Excel* dapat dilihat pada Gambar 4.15.

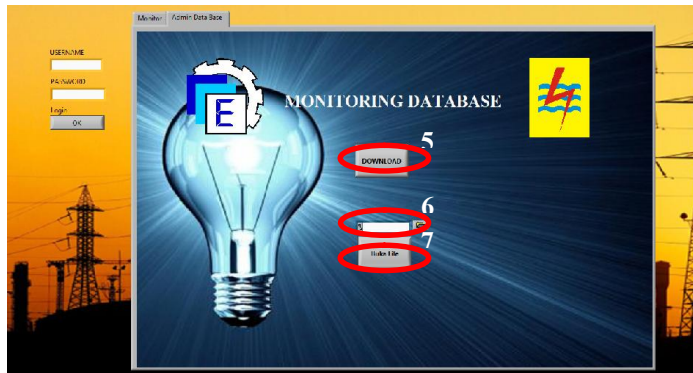


**Gambar 4.13** Tampilan HMI untuk *Admin Database*

Keterangan Gambar 4.13 :

4. Klik *Admin Database* untuk menampilkan HMI *Admin Database*





**Gambar 4.14** Tampilan Untuk *Download Database*

Keterangan Gambar 4.14 :

5. Klik *download* untuk men-*download database* pada HMI *Admin Database*
6. *Cari file*
7. Klik *buka file* untuk membuka *file database* pada HMI *Admin Database*

	A	B	C	D	E
1	Time	Tinggi Air (Cm)	Sensor Tandon	Pompa 1	Pompa 2
2	7/1/2015 23:25:15,503	2	1	1	0
3	7/1/2015 23:26:15,706	5	1	1	0
4	7/1/2015 23:27:15,950	8	1	1	0
5	7/1/2015 23:27:50,030	11	1	1	0
6	7/1/2015 23:28:20,229	11	1	1	0
7	7/1/2015 23:28:50,434	12	1	1	0
8	7/1/2015 23:29:20,579	13	1	1	1
9	7/1/2015 23:30:20,641	6	1	1	1
10	7/1/2015 23:31:20,646	5	1	1	0
11	7/1/2015 23:32:45,789	8	1	1	0
12	7/1/2015 23:33:05,351	11	1	1	0
13	7/1/2015 23:33:35,609	12	1	1	0
14	7/1/2015 23:34:05,919	13	1	1	0
15	7/1/2015 23:34:36,218	12	1	1	1
16	7/1/2015 23:35:36,440	5	1	1	1
17	7/1/2015 23:36:36,782	4	1	1	0
18	7/1/2015 23:37:37,007	8	1	1	0
19	7/1/2015 23:38:15,433	11	1	1	0
20	7/1/2015 23:38:45,755	12	1	1	0
21	7/1/2015 23:39:15,872	13	1	1	0
22	7/1/2015 23:39:52,536	11	1	1	1
23	7/1/2015 23:40:52,919	4	1	1	1
24	7/1/2015 23:41:53,275	6	1	1	0
25	7/1/2015 23:42:53,660	7	1	1	0
26	7/1/2015 23:43:53,897	0	1	1	0

**Gambar 4.15** Tampilan *Database* dalam *Ms.Excel*



#### 4.7 Pengujian Alat Keseluruhan

Pengujian sistem secara keseluruhan merupakan gabungan dari pengujian unit yang saling berhubungan. Pengujian ini dilakukan untuk mengetahui integritas antar unit dalam satu sistem dengan tujuan sistem yang dibangun sesuai dengan perancangan sistem. Pengujian keseluruhan meliputi pengujian LCD, pengujian sensor, pengujian komunikasi, dan pengujian HMI. Monitoring ini dilakukan pada musim penghujan dan banjir di gardu induk Karena lokasi gardu induk lebih rendah dari jalan raya. Hal ini terjadi pada Gardu Induk Sukolilo dan GIS Sawahan. Apabila kabel terendam air dalam waktu yang lama akan terjadi rembesan kedalam inti/tembaga kabel, jika hal tersebut terjadi potensi aliran listrik mengalir pada air menjadi besar.

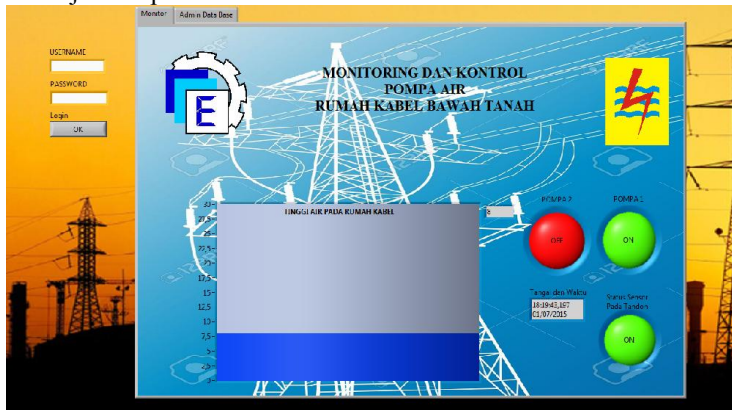
Pengujian dimulai dengan menyalakan sistem dan melihat tampilan LCD apakah berfungsi dengan baik atau tidak. Ketika ada permasalahan pada sistem, pada LCD akan muncul tampilan acak. Sedangkan apabila sistem dapat bekerja dengan baik maka LCD akan menunjukkan tampilan awal ditunjukkan pada Gambar 4.16.



**Gambar 4.16** Tampilan Awal LCD

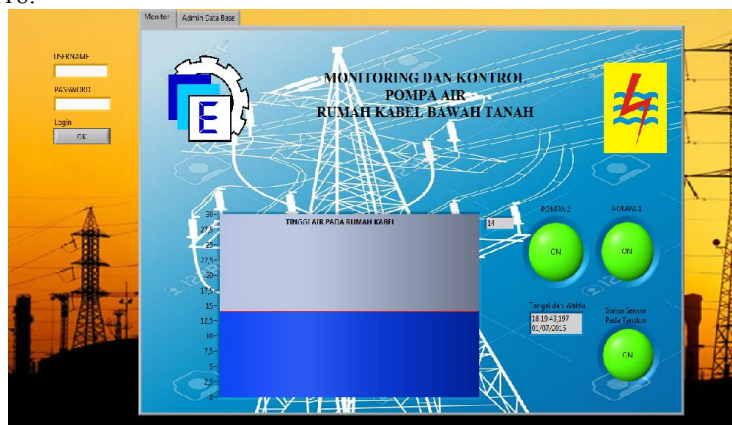
Selanjutnya menguji apakah sensor dapat bekerja dengan baik atau tidak yakni dapat mengukur ketinggian air pada rumah kabel, menguji apakah pompa dapat bekerja dengan baik atau tidak, serta menguji apakah HMI sudah dapat menerima data. dengan baik. Pengujian dilakukan dalam dua kondisi yakni saat kondisi ketinggian air antara 3cm - 13 cm serta saat kondisi ketinggian air diatas 13cm. Pada kondisi awal ketika nilai ketinggian air 8 cm , maka pompa 2 mati, status tandon *on* dan pompa 1 menyala . Pada kondisi ini HMI

menampilkan nilai ketinggian air dan status dari pompa 2 *off*. Kondisi ini ditunjukkan pada Gambar 4.17



**Gambar 4.17** Kondisi Ketinggian 8 cm

Pengujian selanjutnya ialah pengujian dalam kondisi ketinggian air diatas 13 cm. Pada kondisi kedua ini ketika nilai ketinggian air diatas 13 cm, maka pompa 2 akan menyala, status tandon *on* dan pompa 1 menyala. Pada kondisi ini HMI menampilkan nilai ketinggian air dan status dari pompa 1 menyala. Kondisi awal ini ditunjukkan pada Gambar 4.18.



**Gambar 4.18** Kondisi Kedua Ketinggian Diatas 13cm

Dari pengujian sistem secara keseluruhan, semua fungsi sistem dapat bekerja. Sensor mampu merespon terhadap perubahan kondisi lingkungan seperti yang dilakukan dalam prosedur pengujian dan fungsi pompa berjalan dengan normal, LCD dapat menampilkan hasil pembacaan sensor, serta data dapat dikirimkan ke *server* dan HMI dapat menampilkan data sensor. Dengan demikian dapat disimpulkan bahwa alat dapat berfungsi normal.

## **BAB V**

### **PENUTUP**

Setelah melakukan perencanaan dan pembuatan alat serta pengujian dan analisis, maka dapat ditarik kesimpulan dan saran dari kegiatan yang telah dilakukan.

#### **5.1 Kesimpulan**

Dari Tugas Akhir yang telah dikerjakan, dapat disimpulkan bahwa:

1. Alat monitoring dan kontrol pompa pada rumah kabel bawah tanah ini dibatasi hanya memonitoring ketinggian air pada rumah kabel.
2. Pada purwarupa alat ini komponen elektronik selain 2 pompa diwakili oleh lampu pilot untuk indikasi nyala pompa.
3. Sistem kontrol ketinggian air yaitu pada tinggi air lebih dari 13 cm maka pompa 2 *on* dan *off* apabila tinggi air yang semula lebih dari 13 cm menjadi kurang dari 3 cm.
4. Tampilan monitoring pada komputer dapat menampilkan hasil monitoring berupa ketinggian air pada rumah kabel, serta dapat mendeteksi pompa yang bekerja.

#### **5.2 Saran**

Beberapa Dengan memperhatikan beberapa kelemahan dari proyek tugas akhir ini, maka diberikan beberapa saran yang sekiranya dapat dikembangkan pada masa yang akan datang demi kesempurnaan dari proyek tugas akhir ini. Adapun beberapa saran tersebut yaitu:

1. Pengambilan *sample* data sensor dan pengujian alat secara keseluruhan lebih banyak dan berulang, untuk memastikan kemampuan alat serta keakuratan alat.
2. Memakai perangkat elektronik yang lebih kuat dan tahan lama, serta saat pemasangan harus diperhatikan bahwa *grounding* baik.
3. Dalam memonitoring data yang ada sebaiknya dilakukan secara bertahap dengan rentang waktu yang stabil.
4. Pada pengiriman data menggunakan kabel *ethernet*. Untuk mengembangkan tersebut bisa menggunakan SCADA.

*Halaman ini sengaja dikosongkan*

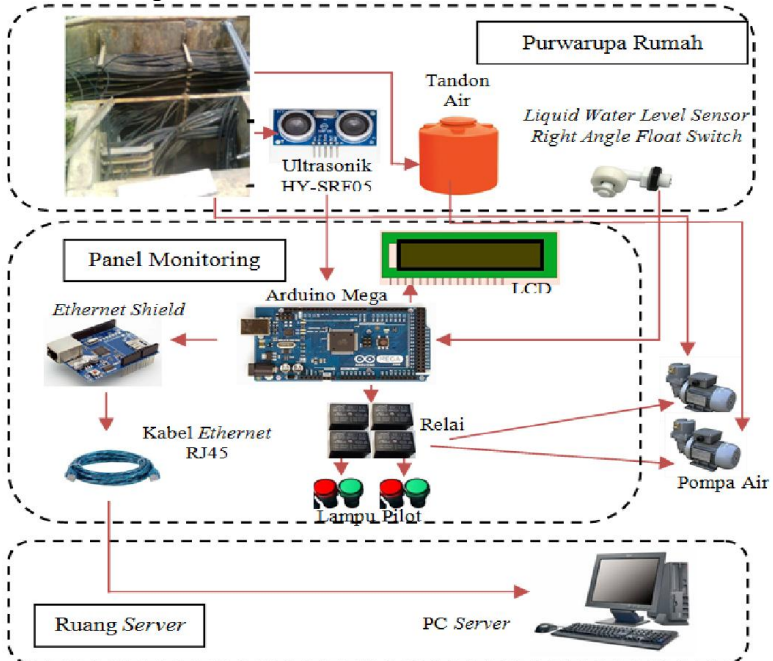
## DAFTAR PUSTAKA

- [1] Bengkel C, *Kabel Tanah Tegangan Menengah 20 kV*, 2013
- [2] Cintia Pratiwi, Shely., “Prototipe Monitoring Kinerja Pompa Dengan Media GSM”, *Tugas Akhir*, D3 Teknik Elektro, ITS Surabaya, 2014.
- [3] .....*Ultrasonik Distance Sensor HY-SRF05*, sensor ultrasonik HY-SRF05, <http://www.pridopia.co.uk/pi-sonic-level.html> (diakses pada tanggal 20 Mei 2015)
- [4] .....*Water Level Sensor Right Angle Float Switch Data Sheet*, Digiware 2015
- [5] Firmansyah, Robbi., “Perancangan Sistem Monitoring Kinerja Ruang Panel Remote Terminal Unit Menggunakan Arduino”, *Tugas Akhir*, D3 Teknik Elektro, ITS Surabaya, 2014.
- [6] Muhammad Syahwil, *Panduan Mudah Simulasi & Praktek Mikrokontroler Arduino*, Penerbit Andi, Jogjakarta, 2013.
- [7] Mubarak, Zakki., “Prototipe Pemutus Saluran Rumah Golongan Pelanggan Kecil Menggunakan Mikrokontroler Dengan Media Wifi”, *Tugas Akhir*, D3 Teknik Elektro, ITS Surabaya, 2014.
- [8] Dian Artanto, *Interaksi Arduino dan LabVIEW*, penerbit Elex Media Komputindo, Jakarta, 2012.
- [9] Ladyada. 2014, *I2C/SPI LCD Backpack*, Adafruit Industries.

*Halaman ini sengaja dikosongkan*

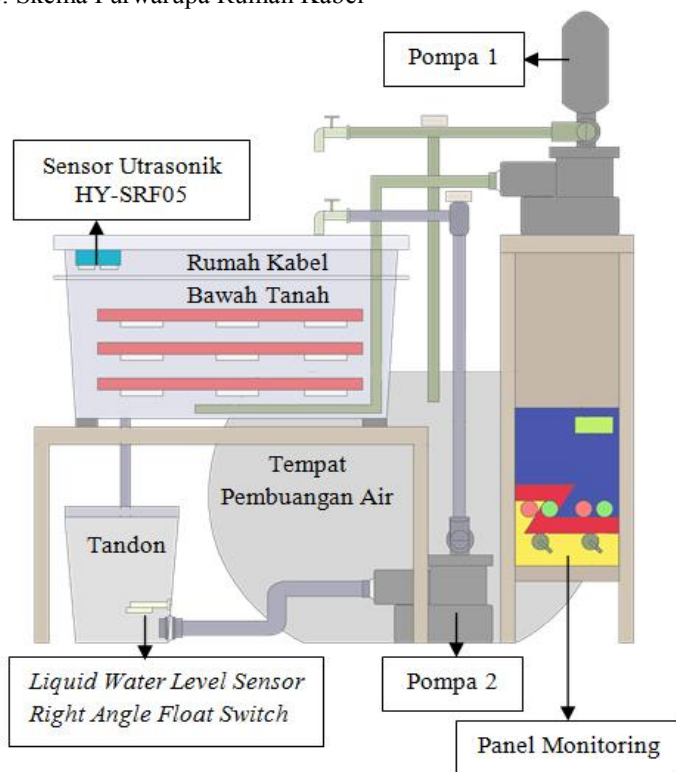
## Lampiran 1 Skema Alat

### a. Perancangan Sistem

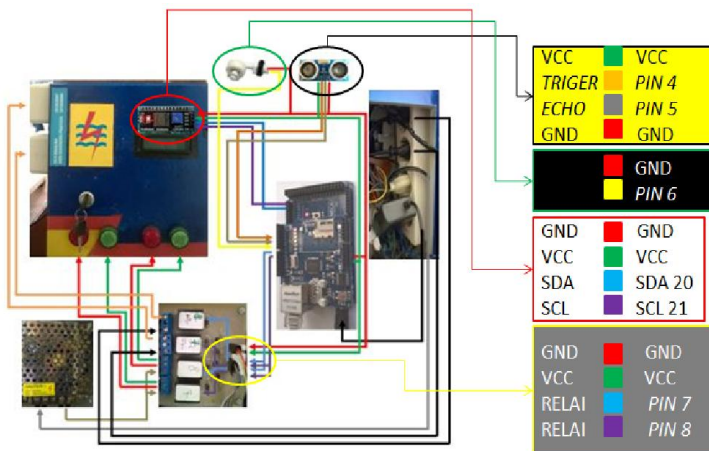
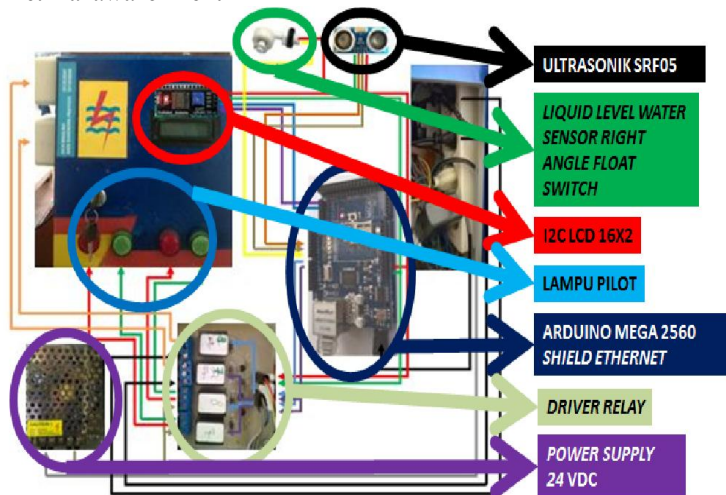




b. Skema Purwarupa Rumah Kabel



c. *Hardware Elektrik*



*Halaman ini sengaja di kosongkan*

## Lampiran 2

*Listing Program pada Arduino*

```
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
#include <DistanceSRF04.h>
#include <stdio.h>
#include <SPI.h>      // needed for Arduino versions later than 0018
#include <Ethernet.h>
#include <EthernetUdp.h>    // UDP library from: bjoern@cs.stanford.edu
12/30/2008
#include <stdio.h>
////////////////////Ethernet
byte mac[] = {
  0xDE, 0xAD, 0xBE, 0xEF, 0xFE, 0xED
};
IPAddress ip(192, 168, 1, 46);

unsigned int localPort = 8888;  // local port to listen on

// buffers for receiving and sending data
char packetBuffer[UDP_TX_PACKET_MAX_SIZE]; //buffer to hold incoming
packet,
char ReplyBuffer[] = "Success";    // a string to send back
int data_masuk;
// An EthernetUDP instance to let us send and receive packets over UDP
EthernetUDP Udp;

////////////////////

LiquidCrystal_I2C lcd(0x27, 2, 1, 0, 4, 5, 6, 7, 3, POSITIVE); // Set the LCD
I2C address
DistanceSRF04 Dist;
int distance;
char data[16];
char ether_jarak[20];
const int buttonPin = 2;  // Button Kanan
const int buttonPin2 = 3;  // Button Kiri
const int ledPin = 8;    // Lampu Kanan
const int ledPin2 = 7;   // Lampu Kiri
const int sensor = 6;
```

```

// variables will change:
int buttonState = 0;
int buttonState2 = 0;
int buttonState3 = 0;
int data_sensor;

int tinggi_air=0,status_air=0;

void setup() {
  // initialize the LED pin as an output:
  Ethernet.begin(mac, ip);
  Udp.begin(localPort);
  Serial.begin(9600);
  Dist.begin(5,4);
  pinMode(ledPin, OUTPUT);
  pinMode(ledPin2, OUTPUT);
  // initialize the pushbutton pin as an input:
  pinMode(buttonPin, INPUT);
  digitalWrite(buttonPin, HIGH);
  pinMode(buttonPin2, INPUT);
  digitalWrite(buttonPin2, HIGH);
  pinMode(sensor, INPUT);
  digitalWrite(sensor, HIGH);
  lcd.begin(16,2);
  lcd.backlight();
}

void loop() {
  // read the state of the pushbutton value:
  buttonState = digitalRead(buttonPin);
  buttonState2 = digitalRead(buttonPin2);
  buttonState3 = digitalRead(sensor);

  distance = Dist.getDistanceCentimeter();
  distance= 27 - distance;
  if (distance <0)
  {
    distance=0;
  }
  // check if the pushbutton is pressed.
  // if it is, the buttonState is HIGH:

```

```

if (buttonState == LOW)
{
    // turn LED on:
    tinggi_air=1;
    digitalWrite(ledPin, HIGH);
}
else
{
    if (tinggi_air==1)
    {
        digitalWrite(ledPin, HIGH);
    }
    else
    {
        // turn LED off:
        digitalWrite(ledPin, LOW);
    }
}
if (buttonState2 == LOW)
{
    // turn LED on:
    digitalWrite(ledPin2, HIGH);
}
else
{
    if (status_air==1)
    {
        digitalWrite(ledPin2, HIGH);
    }
    else
    {
        // turn LED off:
        digitalWrite(ledPin2, LOW);
    }
}
if (buttonState3 == LOW )
{
    // turn LED on:
    data_sensor=1;
    status_air=1;
}
else

```

```

{
  if (buttonState2 == LOW)
  {
    status_air=1;
  }
  else
  {
    // turn LED off:
    data_sensor=0;
    status_air=0;
  }
}
if (tinggi_air==1)
{
  if (distance < 3)
  {
    if (buttonState == LOW)
    {
      tinggi_air =1;
    }
    else
    {
      tinggi_air =0;
    }
  }
}
else if(tinggi_air ==0)
{
  if (distance > 13 )
  {
    tinggi_air =1;
  }
}

//////////////////////////////////////////LCD
if (tinggi_air==1)
{
  if (status_air==0)
  {
    lcd.setCursor(0,0);
    sprintf(data,"P1= Off P2= On  ");
  }
}

```

```

        lcd.print(data);
    }
    else
    {
        lcd.setCursor(0,0);
        sprintf(data,"P1= On P2= On   ");
        lcd.print(data);
    }
}
else
{
    if (status_air==0)
    {
        lcd.setCursor(0,0);
        sprintf(data,"P1= Off P2= Off   ");
        lcd.print(data);
    }
    else
    {
        lcd.setCursor(0,0);
        sprintf(data,"P1= On P2= Off   ");
        lcd.print(data);
    }
}
}
////////////////////////////////////

```

```

Serial.print("\nDistance in centimeters: ");
Serial.print(distance);
Serial.print("    status sensor");
Serial.println(data_sensor);
// lcd.setCursor(0,0); //Start at character 4 on line 0
// lcd.print("Hello, world!");

```

```

lcd.setCursor(0,1);
sprintf(data,"Tinggi Air= %d    ",distance);
lcd.print(data); // Print text on second line
////////////////////////////////////Ethernet
int packetSize = Udp.parsePacket();
if (packetSize)
{
    Serial.print("Received packet of size ");

```



```

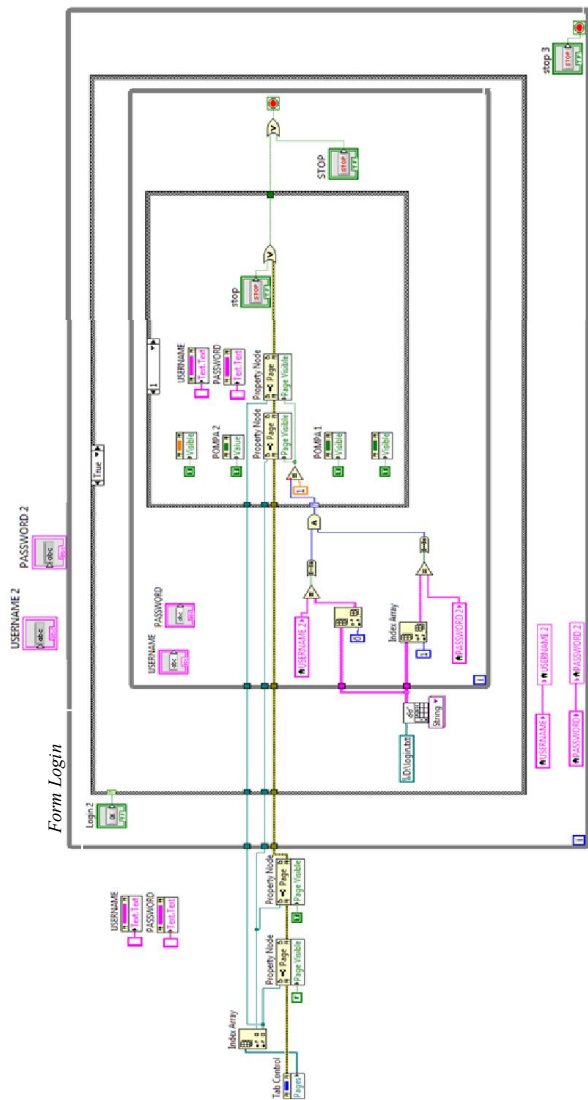
Serial.println(packetSize);
Serial.print("From ");
IPAddress remote = Udp.remoteIP();
for (int i = 0; i < 4; i++)
{
  Serial.print(remote[i], DEC);
  if (i < 3)
  {
    Serial.print(".");
  }
}
Serial.print(", port ");
Serial.println(Udp.remotePort());

// read the packet into packetBuffer
Udp.read(packetBuffer, UDP_TX_PACKET_MAX_SIZE);
Serial.println("Contents:");
Serial.println(packetBuffer);
data_masuk= atoi (packetBuffer);
// send a reply, to the IP address and port that sent us the packet we received
if (data_masuk==1)
{
  sprintf(ether_jarak,"%d",distance);
  Udp.beginPacket(Udp.remoteIP(), Udp.remotePort());
  Udp.write(ether_jarak);
  Udp.endPacket();
}
else if (data_masuk==2)
{
  sprintf(ether_jarak,"%d",status_air);
  Udp.beginPacket(Udp.remoteIP(), Udp.remotePort());
  Udp.write(ether_jarak);
  Udp.endPacket();
}
else if (data_masuk==3)
{
  sprintf(ether_jarak,"%d",tinggi_air);
  Udp.beginPacket(Udp.remoteIP(), Udp.remotePort());
  Udp.write(ether_jarak);
  Udp.endPacket();
}
}

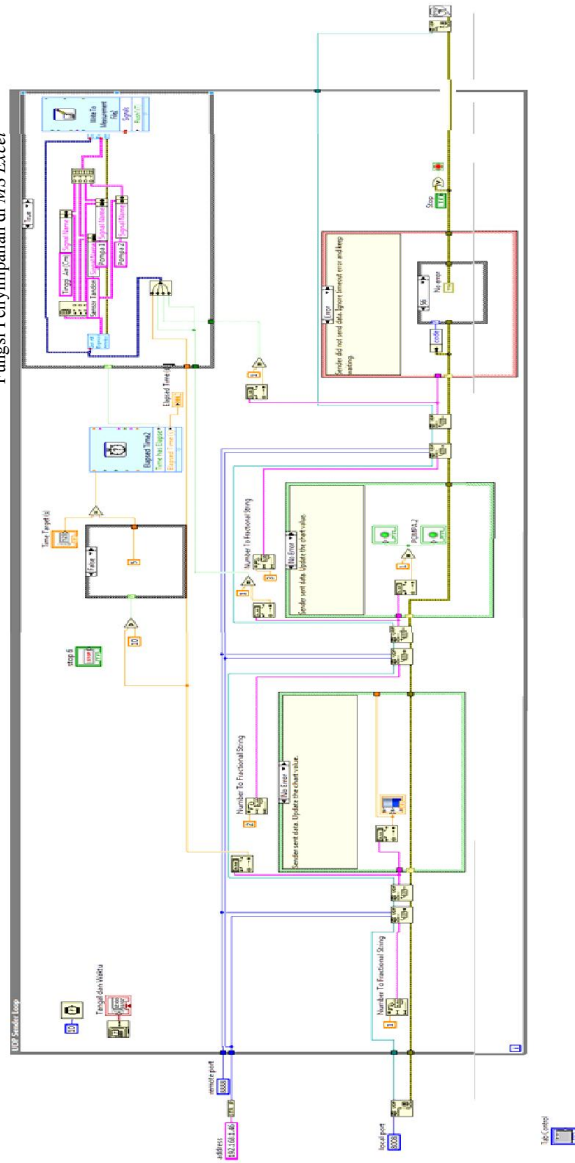
```

```
////////////////////////////////////  
delay(10);  
}
```

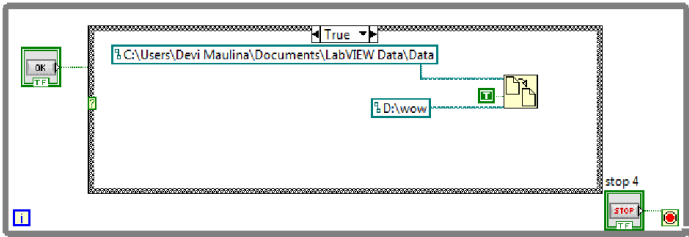
Blok Diagram LabVIEW



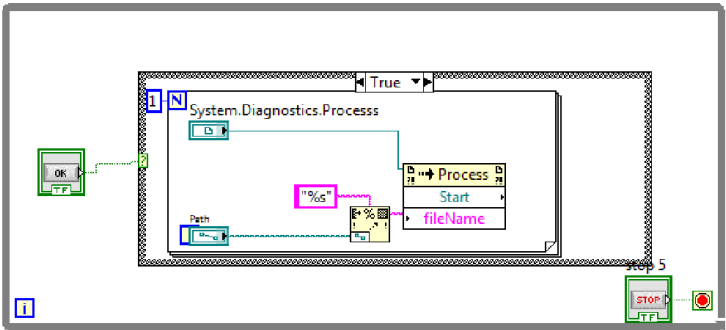
# Fungsi Penyimpanan di MS Excel



Fungsi Meng-copy Database

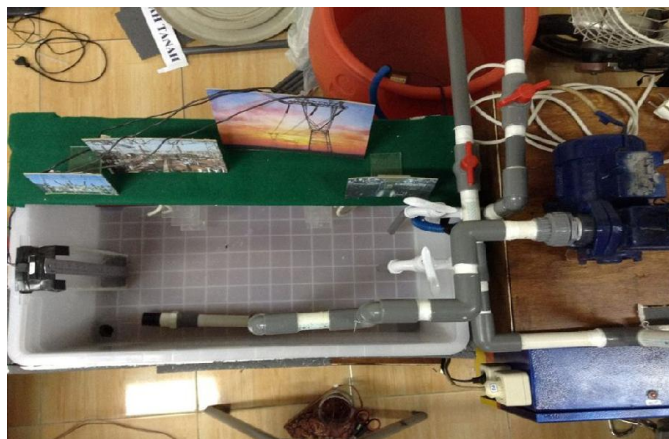


Fungsi Membuka File



### Lampiran 3

Foto Alat Hasil Pembuatan Purwarupa



*Halaman ini sengaja dikosongkan*

## Lampiran 4

### a. Data Sheet Arduino Mega



#### Product Overview

The Arduino Mega 2560 is a microcontroller board based on the ATmega2560 ([datasheet](#)). It has 54 digital input/output pins (of which 14 can be used as PWM outputs), 16 analog inputs, 4 UARTs (hardware serial ports), a 16 MHz crystal oscillator, a USB connection, a power jack, an ICSP header, and a reset button. It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with a AC-to-DC adapter or battery to get started. The Mega is compatible with most shields designed for the Arduino Duemilanove or Diecimila.

#### Index

Technical  
Specifications

Page 2

How to use Arduino  
Programming Enviroment, Basic Tutorials

Page 6

Terms &  
Conditions

Page 7

Enviromental Policies  
half sqm of green via Impatto Zero®

Page 7



# Technical Specification

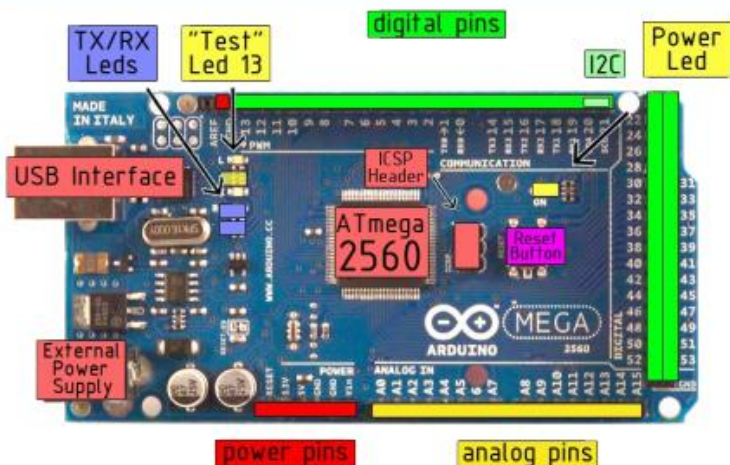


EAGLE files: [arduino-mega2560-reference-design.zip](#) Schematic: [arduino-mega2560-schematic.pdf](#)

## Summary

Microcontroller	ATmega2560
Operating Voltage	5V
Input Voltage (recommended)	7-12V
Input Voltage (limits)	6-20V
Digital I/O Pins	54 (of which 14 provide PWM output)
Analog Input Pins	16
DC Current per I/O Pin	40 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	256 KB of which 8 KB used by bootloader
SRAM	8 KB
EEPROM	4 KB
Clock Speed	16 MHz

## the board



## Power

The Arduino Mega2560 can be powered via the USB connection or with an external power supply. The power source is selected automatically. External (non-USB) power can come either from an AC-to-DC adapter (wall-wart) or battery. The adapter can be connected by plugging a 2.1mm center-positive plug into the board's power jack. Leads from a battery can be inserted in the Gnd and Vin pin headers of the POWER connector.

The board can operate on an external supply of 6 to 20 volts. If supplied with less than 7V, however, the 5V pin may supply less than five volts and the board may be unstable. If using more than 12V, the voltage regulator may overheat and damage the board. The recommended range is 7 to 12 volts.

The Mega2560 differs from all preceding boards in that it does not use the FTDI USB-to-serial driver chip. Instead, it features the Atmega8U2 programmed as a USB-to-serial converter.

The power pins are as follows:

- **VIN.** The input voltage to the Arduino board when it's using an external power source (as opposed to 5 volts from the USB connection or other regulated power source). You can supply voltage through this pin, or, if supplying voltage via the power jack, access it through this pin.
- **5V.** The regulated power supply used to power the microcontroller and other components on the board. This can come either from VIN via an on-board regulator, or be supplied by USB or another regulated 5V supply.
- **3V3.** A 3.3 volt supply generated by the on-board regulator. Maximum current draw is 50 mA.
- **GND.** Ground pins.

## Memory

The ATmega2560 has 256 KB of flash memory for storing code (of which 8 KB is used for the bootloader), 8 KB of SRAM and 4 KB of EEPROM (which can be read and written with the [EEPROM library](#)).

## Input and Output

Each of the 54 digital pins on the Mega can be used as an input or output, using [pinMode\(\)](#), [digitalWrite\(\)](#), and [digitalRead\(\)](#) functions. They operate at 5 volts. Each pin can provide or receive a maximum of 40 mA and has an internal pull-up resistor (disconnected by default) of 20-50 kOhms. In addition, some pins have specialized functions:

- **Serial:** 0 (RX) and 1 (TX); Serial 1: 19 (RX) and 18 (TX); Serial 2: 17 (RX) and 16 (TX); Serial 3: 15 (RX) and 14 (TX). Used to receive (RX) and transmit (TX) TTL serial data. Pins 0 and 1 are also connected to the corresponding pins of the ATmega8U2 USB-to-TTL Serial chip.
- **External Interrupts:** 2 (interrupt 0), 3 (interrupt 1), 18 (interrupt 5), 19 (interrupt 4), 20 (interrupt 3), and 21 (interrupt 2). These pins can be configured to trigger an interrupt on a low value, a rising or falling edge, or a change in value. See the [attachInterrupt\(\)](#) function for details.
- **PWM:** 0 to 13. Provide 8-bit PWM output with the [analogWrite\(\)](#) function.
- **SPI:** 50 (MISO), 51 (MOSI), 52 (SCK), 53 (SS). These pins support SPI communication, which, although provided by the underlying hardware, is not currently included in the Arduino language. The SPI pins are also broken out on the ICSP header, which is physically compatible with the Duemilanove and Diecimila.
- **LED:** 13. There is a built-in LED connected to digital pin 13. When the pin is HIGH value, the LED is on, when the pin is LOW, it's off.
- **I<sup>2</sup>C:** 20 (SDA) and 21 (SCL). Support I<sup>2</sup>C (TWI) communication using the [Wire library](#) (documentation on the Wiring website). Note that these pins are not in the same location as the I<sup>2</sup>C pins on the Duemilanove.

The Mega2560 has 16 analog inputs, each of which provide 10 bits of resolution (i.e. 1024 different values). By default they measure from ground to 5 volts, though it is possible to change the upper end of their range using the AREF pin and [analogReference\(\)](#) function.

There are a couple of other pins on the board:

- **AREF.** Reference voltage for the analog inputs. Used with [analogReference\(\)](#).
- **Reset.** Bring this line LOW to reset the microcontroller. Typically used to add a reset button to shields which block the one on the board.

## Communication

The Arduino Mega2560 has a number of facilities for communicating with a computer, another Arduino, or other microcontrollers. The ATmega2560 provides four hardware UARTs for TTL (5V) serial communication. An ATmega8U2 on the board channels one of these over USB and provides a virtual com port to software on the computer (Windows machines will need a .inf file, but OSX and Linux machines will recognize the board as a COM port automatically). The Arduino software includes a serial monitor which allows simple textual data to be sent to and from the board. The RX and TX LEDs on the board will flash when data is being transmitted via the ATmega8U2 chip and USB connection to the computer (but not for serial communication on pins 0 and 1).

A [SoftwareSerial library](#) allows for serial communication on any of the Mega's digital pins.

The ATmega2560 also supports I2C (TWI) and SPI communication. The Arduino software includes a Wire library to simplify use of the I2C bus; see the [documentation on the Wiring website](#) for details. To use the SPI communication, please see the ATmega2560 datasheet.

## Programming

The Arduino Mega2560 can be programmed with the Arduino software ([download](#)). For details, see the [reference](#) and [tutorials](#).

The ATmega2560 on the Arduino Mega comes preburned with a [bootloader](#) that allows you to upload new code to it without the use of an external hardware programmer. It communicates using the original STK500 protocol ([reference](#), [C header files](#)).

You can also bypass the bootloader and program the microcontroller through the ICSP (In-Circuit Serial Programming) header; see [these instructions](#) for details.



## Automatic (Software) Reset

Rather than requiring a physical press of the reset button before an upload, the Arduino Mega2560 is designed in a way that allows it to be reset by software running on a connected computer. One of the hardware flow control lines (DTR) of the ATmega8U2 is connected to the reset line of the ATmega2560 via a 100 nanofarad capacitor. When this line is asserted (taken low), the reset line drops long enough to reset the chip. The Arduino software uses this capability to allow you to upload code by simply pressing the upload button in the Arduino environment. This means that the bootloader can have a shorter timeout, as the lowering of DTR can be well-coordinated with the start of the upload.

This setup has other implications. When the Mega2560 is connected to either a computer running Mac OS X or Linux, it resets each time a connection is made to it from software (via USB). For the following half-second or so, the bootloader is running on the Mega2560. While it is programmed to ignore malformed data (i.e. anything besides an upload of new code), it will intercept the first few bytes of data sent to the board after a connection is opened. If a sketch running on the board receives one-time configuration or other data when it first starts, make sure that the software with which it communicates waits a second after opening the connection and before sending this data.

The Mega contains a trace that can be cut to disable the auto-reset. The pads on either side of the trace can be soldered together to re-enable it. It's labeled "RESET-EN". You may also be able to disable the auto-reset by connecting a 110 ohm resistor from 5V to the reset line; see [this forum thread](#) for details.

## USB Overcurrent Protection

The Arduino Mega has a resettable polyfuse that protects your computer's USB ports from shorts and overcurrent. Although most computers provide their own internal protection, the fuse provides an extra layer of protection. If more than 500 mA is applied to the USB port, the fuse will automatically break the connection until the short or overload is removed.

## Physical Characteristics and Shield Compatibility

The maximum length and width of the Mega PCB are 4 and 2.1 inches respectively, with the USB connector and power jack extending beyond the former dimension. Three screw holes allow the board to be attached to a surface or case. Note that the distance between digital pins 7 and 8 is 160 mil (0.16"), not an even multiple of the 100 mil spacing of the other pins.

The Mega is designed to be compatible with most shields designed for the Diecimila or Duemilanove. Digital pins 0 to 13 (and the adjacent AREF and GND pins), analog inputs 0 to 5, the power header, and ICSP header are all in equivalent locations. Further the main UART (serial port) is located on the same pins (0 and 1), as are external interrupts 0 and 1 (pins 2 and 3 respectively). SPI is available through the ICSP header on both the Mega and Duemilanove / Diecimila. **Please note that IC is not located on the same pins on the Mega (20 and 21) as the Duemilanove / Diecimila (analog inputs 4 and 5).**



# How to use Arduino



Arduino can sense the environment by receiving input from a variety of sensors and can affect its surroundings by controlling lights, motors, and other actuators. The microcontroller on the board is programmed using the [Arduino programming language](#) (based on [Wiring](#)) and the Arduino development environment (based on [Processing](#)). Arduino projects can be stand-alone or they can communicate with software on running on a computer (e.g. Flash, Processing, MaxMSP).

Arduino is a cross-platform program. You'll have to follow different instructions for your personal OS. Check on the [Arduino site](#) for the latest instructions. <http://arduino.cc/en/Guide/HomePage>

## Linux Install

## Windows Install

## Mac Install

Once you have downloaded/unzipped the arduino IDE, you can Plug the Arduino to your PC via USB cable.

## Blink led

Now you're actually ready to "burn" your first program on the arduino board. To select "blink led", the physical translation of the well known programming "hello world", select

**File>Sketchbook>  
Arduino-0017>Examples>  
Digital>Blink**

Once you have your skecth you'll see something very close to the screenshot on the right.

In **Tools>Board** select MEGA

Now you have to go to **Tools>SerialPort** and select the right serial port, the one arduino is attached to.



Done compiling

Press Compile button  
(to check for errors)



Upload



TX RX Flashing



Blinking Led!



# Terms & Conditions



## 1. Warranties

1.1 The producer warrants that its products will conform to the Specifications. This warranty lasts for one (1) years from the date of the sale. The producer shall not be liable for any defects that are caused by neglect, misuse or mistreatment by the Customer, including improper installation or testing, or for any products that have been altered or modified in any way by a Customer. Moreover, The producer shall not be liable for any defects that result from Customer's design, specifications or instructions for such products. Testing and other quality control techniques are used to the extent the producer deems necessary.

1.2 If any products fail to conform to the warranty set forth above, the producer's sole liability shall be to replace such products. The producer's liability shall be limited to products that are determined by the producer not to conform to such warranty. If the producer elects to replace such products, the producer shall have a reasonable time to replacements. Replaced products shall be warranted for a new full warranty period.

1.3 EXCEPT AS SET FORTH ABOVE, PRODUCTS ARE PROVIDED "AS IS" AND "WITH ALL FAULTS." THE PRODUCER DISCLAIMS ALL OTHER WARRANTIES, EXPRESS OR IMPLIED, REGARDING PRODUCTS, INCLUDING BUT NOT LIMITED TO, ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

1.4 Customer agrees that prior to using any systems that include the producer products, Customer will test such systems and the functionality of the products as used in such systems. The producer may provide technical, applications or design advice, quality characterization, reliability data or other services. Customer acknowledges and agrees that providing these services shall not expand or otherwise alter the producer's warranties, as set forth above, and no additional obligations or liabilities shall arise from the producer providing such services.

1.5 The Arduino<sup>TM</sup> products are not authorized for use in safety-critical applications where a failure of the product would reasonably be expected to cause severe personal injury or death. Safety-Critical Applications include, without limitation, life support devices and systems, equipment or systems for the operation of nuclear facilities and weapons systems. Arduino<sup>TM</sup> products are neither designed nor intended for use in military or aerospace applications or environments and for automotive applications or environment. Customer acknowledges and agrees that any such use of Arduino<sup>TM</sup> products which is solely at the Customer's risk, and that Customer is solely responsible for compliance with all legal and regulatory requirements in connection with such use.

1.6 Customer acknowledges and agrees that it is solely responsible for compliance with all legal, regulatory and safety-related requirements concerning its products and any use of Arduino<sup>TM</sup> products in Customer's applications, notwithstanding any applications-related information or support that may be provided by the producer.

## 2. Indemnification

The Customer acknowledges and agrees to defend, indemnify and hold harmless the producer from and against any and all third-party losses, damages, liabilities and expenses it incurs to the extent directly caused by: (i) an actual breach by a Customer of the representation and warranties made under this terms and conditions or (ii) the gross negligence or willful misconduct by the Customer.

## 3. Consequential Damages Waiver

In no event the producer shall be liable to the Customer or any third parties for any special, collateral, indirect, punitive, incidental, consequential or exemplary damages in connection with or arising out of the products provided hereunder, regardless of whether the producer has been advised of the possibility of such damages. This section will survive the termination of the warranty period.

## 4. Changes to specifications

The producer may make changes to specifications and product descriptions at any time, without notice. The Customer must not rely on the absence or characteristics of any features or instructions marked "reserved" or "undefined." The producer reserves these for future definition and shall have no responsibility whatsoever for conflicts or incompatibilities arising from future changes to them. The product information on the Web Site or Materials is subject to change without notice. Do not finalize a design with this information.

b. *Data Sheet* Sensor Ultrasonik HY-SRF05

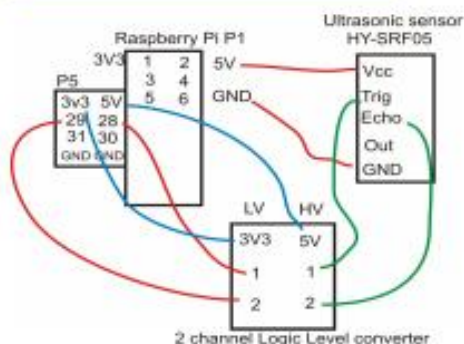
### **Ultrasonic distance sensor (HY-SRF05) with 2channel Logic Level converter**

Use two GPIO Trig(T) Echo(E)

Update --> check distance and update

T28 --> Trig use GPIO28

E29 --> Echo use GPIO29



#### **Ultrasonic sensor module Description :**

- Working Voltage : 5V(DC)
- Static current: Less than 2mA.
- Output signal: Electric frequency signal, high level 5V, low level 0V.
- Sensor angle: Not more than 15 degrees.



- Detection distance: 2cm-450cm.
- High precision: Up to 0.2cm
- Input trigger signal: 10us TTL impulse
- Echo signal : output TTL PWL signal
- Mode of connection:
- 1.VCC    2.trig(T)    3.echo(R)    4.OUT    5.GND
- Use method:
- Supply module with 5V, the output will be 5V while obstacle in range, or 0V if not. The out pin of this module is used as a switching output when anti-theft module, and without the feet when ranging modules.

## 2 channel Logic Level Converter Description :

The logic level converter is a small PCB which safely allows you to interface devices which steps down 5V signals to 3.3V signals and step up 3.3V signals to 5V.

The converter has 2, inputs and outputs and can be used with I2C, SPI, UART, etc.

The board needs to be powered via two sources, 5V and 3.3V and your data lines are linked via pins 1 to 2 on each side so an input on pin 1 will be converted to the lower or higher output on the adjoining pin 1

Download GPIO library

<https://pypi.python.org/pypi/RPi.GPIO>    GPIO library

GPIO library - RPi.GPIO-0.5.3a.tar.gz

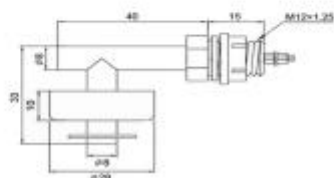
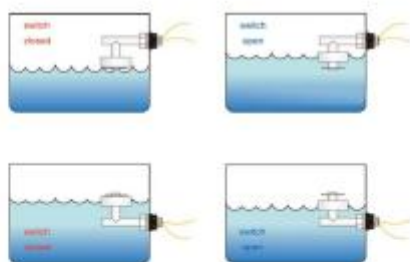
## Install python , library and run the test program

```
# sudo apt-get install python-dev
# wget http://www.pridopia.co.uk/pi-pgm/RPi.GPIO-0.5.3a.tar.gz
# gunzip RPi.GPIO-0.5.3a.tar.gz
# tar -xvf RPi.GPIO-0.5.3a.tar
# cd RPi.GPIO-0.5.3a
# sudo python setup.py install
```

*c. Data Sheet Liquid Water Level Sensor Right Angle Float Switch*

**Liquid Water Level Sensor Right Angle Float Switch**

Product Name	Liquid Water Level Sensor Right Angle Float Switch
Max Contact Rating	10w
Max Switching Voltage	100V DC/AC
Max Switching Current	0.5A
Max Breakdown Voltage	220V DC/AC
Max Load Current	1.0A
Max Contact Resistance	100 mΩ
Temp Rating	-10~+85℃
Float Body Material	P.P
Float Ball Material	P.P
Net Weight	20g/0.7oz



## **LAMPIRAN E**

### **RIWAYAT HIDUP PENULIS**



Nama : Devi Maulina  
NRP : 2212038007  
Jenis Kelamin : Perempuan  
TTL : Surabaya, 6 September 1994  
Alamat Asal : Pondok Benowo Indah EN1  
Alamat Surabaya : Pondok Benowo Indah EN1  
No. HP : 083857323987  
Alamat email : devimaulin@yahoo.co.id

#### **RIWAYAT PENDIDIKAN**

2000 – 2006 : SDN Manukan Kulon V/542 Surabaya  
2006 – 2009 : SMP Ta'miriyah Surabaya  
2009 – 2012 : SMAN 6 Surabaya  
2012 – 2015 : D3 Teknik Elektro ITS Surabaya

#### **PENGALAMAN ORGANISASI**

1. Sekretaris Departemen Sosmas HIMAD3TEKTRO 2013/2014

*Halaman ini sengaja dikosongkan*

## **RIWAYAT HIDUP PENULIS**



Nama : Andi Suhendra Prayuda  
TTL : Sumenep, 24 Januari 1994  
Jenis Kelamin : Laki - Laki  
Agama : Islam  
Alamat Rumah : Perumahan Batuan Blok. ii  
No. 15, Kab. Sumenep  
Telp/HP : 087850367794  
E-mail : asprayuda@gmail.com  
Hobi : Olahraga

### **RIWAYAT PENDIDIKAN**

2000 – 2006 : SDN Batuan 1 Sumenep  
2006 – 2009 : SMPN 4 Sumenep  
2009 – 2012 : SMAN 2 Sumenep  
2012 – 2015 : D3 Teknik Elektro ITS Surabaya

### **PENGALAMAN ORGANISASI**

1. Wakil Ketua Forum Mahasiswa Sumenep di ITS 2014 - 2015
2. Bendahara Perguruan Pencak Silat Anak Bangsa di Sumenep 2011 - 2015

*Halaman ini sengaja dikosongkan*